

09/672 033

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出 願 年 月 日
Date of Application:

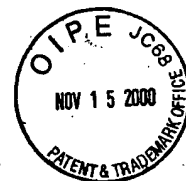
1999年 9月30日

出 願 番 号
Application Number:

平成11年特許願第279317号

出 願 人
Applicant (s):

ブラザー工業株式会社

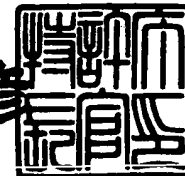


CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 3月24日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3020567

【書類名】 特許願

【整理番号】 99043101

【提出日】 平成11年 9月30日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/00

【発明の名称】 テキストデータ処理装置、テキストデータ処理方法および記録媒体

【請求項の数】 12

【発明者】

 【住所又は居所】 愛知県名古屋市瑞穂区苗代町 1 5 番 1 号 ブラザー工業株式会社内

 【氏名】 桑原 宏和

【発明者】

 【住所又は居所】 愛知県名古屋市瑞穂区苗代町 1 5 番 1 号 ブラザー工業株式会社内

 【氏名】 岩永 和彦

【発明者】

 【住所又は居所】 愛知県名古屋市瑞穂区苗代町 1 5 番 1 号 ブラザー工業株式会社内

 【氏名】 池戸 辰裕

【発明者】

 【住所又は居所】 愛知県名古屋市瑞穂区苗代町 1 5 番 1 号 ブラザー工業株式会社内

 【氏名】 伊藤 千年

【特許出願人】

 【識別番号】 000005267

 【氏名又は名称】 ブラザー工業株式会社

【代理人】

 【識別番号】 100089196

【弁理士】

【氏名又は名称】 梶 良之

【選任した代理人】

【識別番号】 100104226

【弁理士】

【氏名又は名称】 須原 誠

【手数料の表示】

【予納台帳番号】 014731

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 テキストデータ処理装置、テキストデータ処理方法および記録媒体

【特許請求の範囲】

【請求項 1】 テキストデータ中の改行コードを検出する改行コード検出手段と、

前記改行コード検出手段によって改行コードが検出された際に、当該改行コードの直前または直後に所定コードがあるかどうかを検出する所定コード検出手段と、

前記所定コード検出手段によって所定コードが検出されなかった際に、当該改行コードを削除する改行コード削除手段とを備えていることを特徴とするテキストデータ処理装置。

【請求項 2】 テキストデータ中のスペース文字を検出するスペース文字検出手段と、

前記スペース文字検出手段がスペース文字を検出するごとに 1 だけ加算され、前記スペース文字検出手段がスペース文字以外を検出することによって初期化されるスペースカウンタと、

前記スペース文字検出手段によってスペース文字が検出されなかった際に、前記スペースカウンタが 2 以上となっているかを判定するスペース数判定手段と、

前記スペース数判定手段によって前記スペースカウンタが 2 以上となっていると判定された際に、当該 2 以上のスペース文字を削除する複数スペース文字削除手段とを備えていることを特徴とするテキストデータ処理装置。

【請求項 3】 行頭にスペース文字が 1 つだけ存在していることを検出する行頭スペース文字検出手段と、

前記行頭スペース文字検出手段によって行頭にスペース文字が 1 つだけ存在していることが検出された際に、当該 1 つのスペース文字を削除する行頭スペース文字削除手段とを備えていることを特徴とするテキストデータ処理装置。

【請求項 4】 テキストデータ中の改行コードを検出する改行コード検出ステップと、

前記改行コード検出ステップによって改行コードが検出された際に、当該改行コードの直前または直後に所定コードがあるかどうかを検出する所定コード検出ステップと、

前記所定コード検出手段によって所定コードが検出されなかった際に、当該改行コードを削除する改行コード削除ステップとを有していることを特徴とするテキストデータ処理方法。

【請求項 5】 テキストデータ中のスペース文字を検出するスペース文字検出ステップと、

前記スペース文字検出ステップがスペース文字を検出するごとにスペースカウンタを 1 だけ加算するスペースカウンタ加算ステップと、

前記スペース文字検出ステップによってスペース文字が検出されなかった際に、前記スペースカウンタが 2 以上となっているかを判定するスペース数判定ステップと、

前記スペース数判定ステップによって前記スペースカウンタが 2 以上とっていると判定された際に、当該 2 以上のスペース文字を削除する複数スペース文字削除ステップとを有していることを特徴とするテキストデータ処理方法。

【請求項 6】 行頭にスペース文字が 1 つだけ存在していることを検出する行頭スペース文字検出ステップと、

前記行頭スペース文字検出ステップによって行頭にスペース文字が 1 つだけ存在していることが検出された際に、当該 1 つのスペース文字を削除する行頭スペース文字削除ステップとを有していることを特徴とするテキストデータ処理方法。

【請求項 7】 テキストデータ中の改行コードを検出する改行コード検出手段、

前記改行コード検出手段によって改行コードが検出された際に、当該改行コードの直前または直後に所定コードがあるかどうかを検出する所定コード検出手段、および、

前記所定コード検出手段によって所定コードが検出されなかった際に、当該改行コードを削除する改行コード削除手段としてコンピュータを機能させるための

プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 8】 テキストデータ中のスペース文字を検出するスペース文字検出手段、

前記スペース文字検出手段がスペース文字を検出するごとに 1 だけ加算され、前記スペース文字検出手段がスペース文字以外を検出することによって初期化されるスペースカウンタ、

前記スペース文字検出手段によってスペース文字が検出されなかった際に、前記スペースカウンタが 2 以上となっているかを判定するスペース数判定手段、および、

前記スペース数判定手段によって前記スペースカウンタが 2 以上となっていると判定された際に、当該 2 以上のスペース文字を削除する複数スペース文字削除手段としてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 9】 行頭にスペース文字が 1 つだけ存在していることを検出する行頭スペース文字検出手段、および、

前記行頭スペース文字検出手段によって行頭にスペース文字が 1 つだけ存在していることが検出された際に、当該 1 つのスペース文字を削除する行頭スペース文字削除手段としてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 10】 テキストデータ中の改行コードおよびスペース文字を検出する改行コード・スペース文字検出手段と、

前記改行コード・スペース文字検出手段によって改行コードが検出された際に、当該改行コードの直前または直後に所定コードがあるかどうかを検出する所定コード検出手段と、

前記所定コード検出手段によって所定コードが検出されなかった際に、当該改行コードを削除する改行コード削除手段と、

前記改行コード・スペース文字検出手段がスペース文字を検出するごとに 1 だけ加算され、前記改行コード・スペース文字検出手段がスペース文字以外を検出することによって初期化されるスペースカウンタと、

前記改行コード・スペース文字検出手段が改行コードおよびスペース文字以外を検出した際に、前記スペースカウンタが 2 以上となっているかを判定するスペース数判定手段と、

前記スペース数判定手段によって前記スペースカウンタが 2 以上となっていると判定された際に、当該 2 以上のスペース文字を削除する複数スペース文字削除手段とを備えていることを特徴とするテキストデータ処理装置。

【請求項 1 1】 1 つの論理行の改行コードの前にスペース文字以外の文字が存在するかどうかを表す空白行フラグを記憶する空白行フラグ記憶手段と、

前記空白行フラグ記憶手段に記憶された空白行フラグを、初期化時および前記改行コード・スペース文字検出手段が改行コードを検出した際であって前記スペースカウンタが初期化される直前にオンとし、前記改行コード・スペース文字検出手段が改行コードおよびスペース文字以外を検出した際であって前記スペースカウンタが初期化される直前にオフとする空白行フラグ制御手段と、

前記スペース数判定手段によって前記スペースカウンタが 2 以上とっていないと判定された際に、前記空白行フラグ記憶手段に記憶された空白行フラグがオンであれば、当該論理行の行頭にある 1 つのスペース文字を削除する行頭スペース文字削除手段とをさらに備えていることを特徴とする請求項 1 0 に記載のテキストデータ処理装置。

【請求項 1 2】 前記改行コード・スペース文字検出手段によって改行コードが検出された際に、前記空白行フラグ記憶手段に記憶された空白行フラグがオンであれば、当該論理行のすべてのスペース文字を削除する全スペース文字削除手段をさらに備えていることを特徴とする請求項 1 0 または 1 1 に記載のテキストデータ処理装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、テキストデータ処理装置、テキストデータ処理方法および記録媒体に関し、特に、テキストデータを見やすく表示或いは印刷させることが可能なテキストデータ処理装置、テキストデータ処理方法および記録媒体に関する。

【0002】

【従来の技術】

近年のコンピュータ技術の進歩およびインターネットの普及に伴い、ユーザはWWW (World Wide Web) を介して様々な情報に迅速にアクセスし、一般にWWWブラウザ（以下、単に「ブラウザ」という）と称される閲覧ソフトウェアに所望の文字列や画像を手軽に表示させることが可能となっている。ところが、外出先などのコンピュータが使用できない場所でアクセスした情報を参照するためには、これを紙に印刷しておくことが必要な場合も多い。そして、そのためには、ブラウザの印刷機能を用いて、ブラウザに表示された文字列や画像をそのままの状態で、例えばA4サイズなどの紙に印刷するのが最も簡単である。

【0003】

しかしながら、ブラウザに表示された情報のうち実際に必要となる情報はその一部に過ぎないことが多く、ブラウザの印刷機能を用いて印刷すると、本来必要でない情報までも印刷されることになってしまう。そのため、印刷された紙の枚数が増えて重くなり持ち運ぶのに適さない、保存するのに場所をとる、必要な情報を探すのに時間がかかる、用紙を無駄に使用してしまうなどの問題を生じてしまう。

【0004】

これらの問題を回避するには、ブラウザに表示されたデータから必要なデータだけをコピー&ペーストするなどして例えばワープロソフトに取り出し、さらに取り出したデータをワープロソフトの様々な機能を使って好ましいレイアウトに編集するという操作が必要である。

【0005】

【発明が解決しようとする課題】

しかしながら、ブラウザから必要なデータをワープロソフトなどに取り出して編集する場合には、非常に煩雑な作業が必要となる。例えば、ブラウザから取り出されたテキストデータを所望の大きさの用紙に印刷するには、テキストデータが見やすくなるように整形する作業を行うことが好ましい。特にWWWにおいて標準的に用いられているHTML文書から取り込んだテキストデータは、各表示

行ごとに改行コードが挿入されていたり、インデントが多用されていることが多く、これを異なる行内表示文字数で印刷すると行の途中で文章が改行されたり、スペース文字によって行頭位置が不揃いになったりして非常に読みづらい文章となってしまう。ところが、これら改行コードやスペース文字を一律に削除してしまうとかえって文章が読みにくくなってしまい、改行コードやスペース文字を削除するユーザが個別の判断で行うしかなく、これらの作業には多大なる時間と手間が必要である。

【0006】

そこで、本発明の目的は、煩雑な作業を行う必要なく所望サイズの領域に見やすく表示或いは印刷させることができるようにテキストデータを処理するテキストデータ処理装置、テキストデータ処理方法および記録媒体を提供することである。

【0007】

【課題を解決するための手段】

上記目的を達成するために、請求項1のテキストデータ処理装置は、テキストデータ中の改行コードを検出する改行コード検出手段と、前記改行コード検出手段によって改行コードが検出された際に、当該改行コードの直前または直後に所定コードがあるかどうかを検出する所定コード検出手段と、前記所定コード検出手段によって所定コードが検出されなかった際に、当該改行コードを削除する改行コード削除手段とを備えていることを特徴とするものである。

【0008】

請求項4のテキストデータ処理方法は、テキストデータ中の改行コードを検出する改行コード検出ステップと、前記改行コード検出ステップによって改行コードが検出された際に、当該改行コードの直前または直後に所定コードがあるかどうかを検出する所定コード検出ステップと、前記所定コード検出手段によって所定コードが検出されなかった際に、当該改行コードを削除する改行コード削除ステップとを有していることを特徴とするものである。

【0009】

請求項7は、テキストデータ中の改行コードを検出する改行コード検出手段、

前記改行コード検出手段によって改行コードが検出された際に、当該改行コードの直前または直後に所定コードがあるかどうかを検出する所定コード検出手段、および、前記所定コード検出手段によって所定コードが検出されなかった際に、当該改行コードを削除する改行コード削除手段としてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0010】

請求項1、4、7によると、直前または直後に所定コード（例えば、直前にある「。」、「、」、「、」、「:」といった句読点などの区切り記号や、直後にある「・」、「1.」、「a.」などの箇条書きの先頭に通常使用される箇条書き記号など）が存在しない改行コードだけを選択して自動的に削除することができる。つまり、区切り記号直後の改行コードは文章を読みやすくするために文章作成者が意図的に挿入したものであること多く、これを削除するとかえって文章が読みづらくなってしまう。また、箇条書き記号直前の改行コードについても同様である。そこで、これらの改行コードは削除せず、それ以外の改行コードが自動的に削除されるようにすることで、ユーザが個々のテキストデータから不要な改行コードだけを削除するという煩雑な作業を必要とせずに、テキストデータをその行内表示文字数が変わった場合であっても見やすく表示或いは印刷させることが可能となる。

【0011】

なお、具体的にどのような区切り記号および箇条書き記号を採用するかは、予めデフォルトで決めておくことが好ましいが、ユーザがテキストの内容に応じて適宜変更できるようになっていることが好ましい。また、テキストデータとは、文字や記号がShift-JISコードやUNICODEなどの所定コードで表された文字列データであって、txt、html、rtfなどのファイル形式のほかに特定のワープロソフトで書き込まれるファイル形式で表されるデータをも含むものとする。また、表（テーブル）データは、枠線を罫線に置き換えることによってテキストデータとして扱うようにしてもよい。表以外のデータであってテキスト或いは画像のいずれでもないデータについても、適宜テキストデータ或いは画像データのいずれかとして扱えばよい。

【0012】

また、請求項2のテキストデータ処理装置は、テキストデータ中のスペース文字を検出するスペース文字検出手段と、前記スペース文字検出手段がスペース文字を検出するごとに1だけ加算され、前記スペース文字検出手段がスペース文字以外を検出することによって初期化されるスペースカウンタと、前記スペース文字検出手段によってスペース文字が検出されなかった際に、前記スペースカウンタが2以上となっているかを判定するスペース数判定手段と、前記スペース数判定手段によって前記スペースカウンタが2以上となっていると判定された際に、当該2以上のスペース文字を削除する複数スペース文字削除手段とを備えていることを特徴とするものである。

【0013】

請求項5のテキストデータ処理方法は、テキストデータ中のスペース文字を検出するスペース文字検出ステップと、前記スペース文字検出ステップがスペース文字を検出するごとにスペースカウンタを1だけ加算するスペースカウンタ加算ステップと、前記スペース文字検出ステップによってスペース文字が検出されなかった際に、前記スペースカウンタが2以上となっているかを判定するスペース数判定ステップと、前記スペース数判定ステップによって前記スペースカウンタが2以上となっていると判定された際に、当該2以上のスペース文字を削除する複数スペース文字削除ステップとを有していることを特徴とするものである。

【0014】

請求項8は、テキストデータ中のスペース文字を検出するスペース文字検出手段、前記スペース文字検出手段がスペース文字を検出するごとに1だけ加算され、前記スペース文字検出手段がスペース文字以外を検出することによって初期化されるスペースカウンタ、前記スペース文字検出手段によってスペース文字が検出されなかった際に、前記スペースカウンタが2以上となっているかを判定するスペース数判定手段、および、前記スペース数判定手段によって前記スペースカウンタが2以上となっていると判定された際に、当該2以上のスペース文字を削除する複数スペース文字削除手段としてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0015】

請求項2、5、8によると、2以上連続したスペース文字（全角または半角スペース、タブコードなど）を検出してこれを自動的に削除することができるので、ユーザが個々のテキストデータから不要なスペース文字だけを削除するという煩雑な作業を必要とせずに、例えば単語間に故意に挿入された1つのスペース文字を削除することなく、テキストデータをその行内表示文字数が変わった場合であっても見やすく表示或いは印刷させることが可能となる。なお、具体的にどのようなスペース文字を採用するかは、予めデフォルトで決めておくことが好ましいが、ユーザがテキストの内容に応じて適宜変更できるようになっていることが好ましい。

【0016】

また、請求項3のテキストデータ処理装置は、行頭にスペース文字が1つだけ存在していることを検出する行頭スペース文字検出手段と、前記行頭スペース文字検出手段によって行頭にスペース文字が1つだけ存在していることが検出された際に、当該1つのスペース文字を削除する行頭スペース文字削除手段とを備えていることを特徴とするものである。

【0017】

請求項6のテキストデータ処理方法は、行頭にスペース文字が1つだけ存在していることを検出する行頭スペース文字検出ステップと、前記行頭スペース文字検出ステップによって行頭にスペース文字が1つだけ存在していることが検出された際に、当該1つのスペース文字を削除する行頭スペース文字削除ステップとを有していることを特徴とするものである。

【0018】

請求項9は、行頭にスペース文字が1つだけ存在していることを検出する行頭スペース文字検出手段、および、前記行頭スペース文字検出手段によって行頭にスペース文字が1つだけ存在していることが検出された際に、当該1つのスペース文字を削除する行頭スペース文字削除手段としてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0019】

請求項 3、6、9 によると、行頭に 1 つだけ存在するスペース文字が削除されるので、単語間などに故意に挿入された 1 つのスペース文字が削除されないようにしつつ、行頭にあるスペース文字を削除することができる。そのため、行頭に統一感が生み出され、テキストデータが見やすくなる。

【0020】

また、請求項 10 のテキストデータ処理装置は、テキストデータ中の改行コードおよびスペース文字を検出する改行コード・スペース文字検出手段と、前記改行コード・スペース文字検出手段によって改行コードが検出された際に、当該改行コードの直前または直後に所定コードがあるかどうかを検出する所定コード検出手段と、前記所定コード検出手段によって所定コードが検出されなかった際に、当該改行コードを削除する改行コード削除手段と、前記改行コード・スペース文字検出手段がスペース文字を検出するごとに 1 だけ加算され、前記改行コード・スペース文字検出手段がスペース文字以外を検出することによって初期化されるスペースカウンタと、前記改行コード・スペース文字検出手段が改行コードおよびスペース文字以外を検出した際に、前記スペースカウンタが 2 以上となっているかを判定するスペース数判定手段と、前記スペース数判定手段によって前記スペースカウンタが 2 以上となっていると判定された際に、当該 2 以上のスペース文字を削除する複数スペース文字削除手段とを備えていることを特徴とするものである。

【0021】

請求項 10 によると、直前または直後に所定コードが存在しない改行コードだけを選択して自動的に削除することができるとともに、2 以上連続したスペース文字を検出して自動的に削除することができるので、ユーザが個々のテキストデータから不要な改行コードおよびスペース文字だけを削除するという煩雑な作業を必要とせず、テキストデータをその行内表示文字数が変わった場合であっても見やすく表示或いは印刷させることが可能となる。

【0022】

また、請求項 11 のテキストデータ処理装置は、1 つの論理行の改行コードの前にスペース文字以外の文字が存在するかどうかを表す空白行フラグを記憶する

空白行フラグ記憶手段と、前記空白行フラグ記憶手段に記憶された空白行フラグを、初期化時および前記改行コード・スペース文字検出手段が改行コードを検出した際であって前記スペースカウンタが初期化される直前にオンとし、前記改行コード・スペース文字検出手段が改行コードおよびスペース文字以外を検出した際であって前記スペースカウンタが初期化される直前にオフとする空白行フラグ制御手段と、前記スペース数判定手段によって前記スペースカウンタが2以上とになっていないと判定された際に、前記空白行フラグ記憶手段に記憶された空白行フラグがオンであれば、当該論理行の行頭にある1つのスペース文字を削除する行頭スペース文字削除手段とをさらに備えていることを特徴とするものである。

【0023】

請求項11によると、空白行フラグを用いて論理行の行頭にある1つのスペース文字を削除することができるので、単語間などに故意に挿入された1つのスペース文字が削除されないようにしつつ、行頭にあるスペース文字を削除することができる。そのため、行頭に統一感が生み出され、テキストデータが見やすくなる。

【0024】

また、請求項12のテキストデータ処理装置は、前記改行コード・スペース文字検出手段によって改行コードが検出された際に、前記空白行フラグ記憶手段に記憶された空白行フラグがオンであれば、当該論理行のすべてのスペース文字を削除する全スペース文字削除手段をさらに備えていることを特徴とするものである。

【0025】

請求項12によると、空白行フラグを用いて改行コード（または文頭）と改行コードとの間にスペース文字だけがある論理行のすべてのスペース文字を削除することができるので、効率的に不要なスペースを削除してテキストデータをその行内表示文字数が変わった場合であっても見やすく表示或いは印刷させることが可能となる。

【0026】

【発明の実施の形態】

以下、本発明の好適な実施の形態について、図面を参照しつつ説明する。

【0027】

図1は、本実施の形態のテキストデータ処理装置であるパーソナルコンピュータと、これに接続されたシート加工装置とからなるラベル製造装置のブロック図である。ここでは、まず、シート加工装置（以下、「カッティングプリンタ」という）の構造について説明する。図2は、カッティングプリンタの要部構成を示す平面図である。図3は、図2に示すカッティングプリンタの要部断面図である。図4は、図2に示すカッティングプリンタのカッティング機構部を示す要部側断面図である。また、図5は図2に示すロールシートユニットとカッティング機構部および画像形成機構部との概略的な位置関係を示す模式的な斜視図である。

【0028】

図2および図3に示すカッティングプリンタ11は、左右両側に配設される側壁9および10を持つフレーム12内に、シートとしてのタックシート13を巻回した状態で回転自在に支持しているロールシートユニット14と、タックシート13を正逆両方向に搬送し得る搬送手段としての搬送機構部15と、搬送機構部15によって搬送されるタックシート13をカットするためのカッティング機構部16と、タックシート13の正方向（排出方向）への搬送に関してカッティング機構部16の上流側に配設され、タックシート13に所定の画像を形成するための画像形成手段としての画像形成機構部17とを有している。

【0029】

ロールシート51は、図5に示すように、タックシート13が筒状の巻芯55にロール状に巻回されたものである。なお、このタックシート13は、表面が印刷可能とされ裏面に粘着剤が塗布された粘着シート18と、この粘着シート18の裏面に貼り合わされる離型紙19との2層によって構成されている。

【0030】

カッティングプリンタ11を構成する各部分のうち、まず、搬送機構部15について説明する。搬送機構部15は、図3に示すように、画像形成機構部17の構成要素でもあるプラテンローラ24と、カッティング機構部16の下流側に配設される排出ローラ25とを備えている。排出ローラ25とタックシート13を

挟んで対向する位置には、従動ローラ 8 が配設されている。この従動ローラ 8 は、従動ローラ 8 をばねにより排出口ローラ 2 5 側に押圧付勢する押さえ板 7 によって支持されている。そして、フレーム 1 2 内に配設された第 1 駆動モータ 2 1 の正逆駆動によって、プラテンローラ 2 4 および排出口ローラ 2 5 は、第 1 ギヤ列 2 2 を介して、タックシート 1 3 を正方向および逆方向に搬送するように回転させられる。

【0 0 3 1】

また、ロールシートユニット 1 4 のフランジギヤ（図示せず）に設けられたギヤ 5 9 には、遊星ギヤ機構 2 6 を含む第 2 ギヤ列 2 7 を介して第 1 駆動モータ 2 1 からの駆動が伝達されるように構成されている。この遊星ギヤ機構 2 6 は、タックシート 1 3 を逆方向に搬送するときのみギヤ 5 9 と噛み合い、タックシート 1 3 を正方向に搬送するときにはギヤ 5 9 と噛み合わないよう構成されている。したがって、タックシート 1 3 を正方向に搬送するときには、プラテンローラ 2 4 および排出口ローラ 2 5 の回転によってタックシート 1 3 を引き出す力によりロールシート 5 1 が回転し、一方、タックシート 1 3 を逆方向に搬送するときには、第 1 駆動モータ 2 1 からの駆動によってロールシート 5 1 が逆方向に回転する。

【0 0 3 2】

次に、カッティング機構部 1 6 について説明する。カッティング機構部 1 6 は、タックシート 1 3 をその下方において受ける裁断ベッド 2 9 と、この裁断ベッド 2 9 とタックシート 1 3 を挟んで対向するカット手段としてのカタユニット 3 0 と、このカタユニット 3 0 が着脱自在に装着されるキャリッジ 3 1 とを備えている。

【0 0 3 3】

図 4 に示すように、カタユニット 3 0 は、その下端側にタックシート 1 3 をカットするためのカタ 4 3 を有している。カタ 4 3 は、カタユニット 3 0 内の図示しない昇降機構によって、粘着シート 1 8 とともに離型紙 1 9 をもカットする全カット（フルカット）位置、および、粘着シート 1 8 のみをカットするハーフカット位置のいずれかの位置に選択的に支持される。

【 0 0 3 4 】

図 2 に示すように、キャリッジ 3 1 は、フレーム 1 2 の両側壁 9 および 1 0 の外側に配設された一対のプーリ 3 2 および 3 3 に掛け渡されたエンドレス状のタイミングベルト 3 4 の 1 箇所 に連結されている。図 3 に示すように、一対のプーリ 3 2 および 3 3 のうち、1 つのプーリ 3 2 は、フレーム 1 2 の側壁 1 0 の外側に配設された第 2 駆動モータ 3 5 によって、ベベルギヤなどを含む第 3 ギヤ列 3 6 を介して駆動される。これによって、キャリッジ 3 1 は、タックシート 1 3 の搬送方向と略直交する方向（シートの幅方向）に往復移動される。

【 0 0 3 5 】

また、図 4 に示すように、このキャリッジ 3 1 におけるカッタユニット 3 0 が装着される側の端部と反対側の端部には、その両端がフレーム 1 2 の両側壁 9 および 1 0 に支持されたメインガイドシャフト 3 7 が挿通されており、キャリッジ 3 1 はメインガイドシャフト 3 7 に摺動可能に支持されている。また、カッタユニット 3 0 が装着された側の端部とメインガイドシャフト 3 7 が挿通された側の端部との間の途中部位には、メインガイドシャフト 3 7 と略平行に延びる補助ガイドシャフト 3 8 が摺動可能に挿通されている。この補助ガイドシャフト 3 8 の両端部は、フレーム 1 2 の両側壁 9 および 1 0 に回動可能に設けられた一対の回動アーム 3 9 に支持されている。

【 0 0 3 6 】

また、補助ガイドシャフト 3 8 の一方の端部は、作動リンク 4 0 を介してソレノイド 4 1 の出力軸 4 2 に連結されている。そして、ソレノイド 4 1 がオン状態（励磁状態）のときには、図示しないばねの付勢力によって、カッタユニット 3 0 は、その下端側が裁断部ベット 2 9 の上面を押圧するように作動させられる。また、ソレノイド 4 1 がオフ状態（励磁されていない状態）のときには、ソレノイド 4 1 の出力軸 4 2 が上方に進出し、作動リンク 4 0 および補助ガイドシャフト 3 8 を介してキャリッジ 3 1 のカッタユニット 3 0 が装着される側の端部がメインガイドシャフト 3 7 を支点として上向きに回動する。従って、カッタユニット 3 0 は、その下端側が裁断部ベット 2 9 の上面から離れるように作動させられる。

【0037】

次に、画像形成機構部 1 7 について説明する。図 2 および図 3 に示すように、画像形成機構部 1 7 は、タックシート 1 3 の幅寸法にほぼ等しい長さを持つ、印字ヘッドとしてのライン型のサーマルヘッド 4 4 と、このサーマルヘッド 4 4 とタックシート 1 3 を挟んで対向するプラテンローラ 2 4 とを備えている。

【0038】

上述したカッティングプリンタ 1 1 に用いられるタックシート 1 3 は、粘着シート 1 8 と離型紙 1 9 との 2 層によって構成されているので、必要な情報が印刷された際に、離型紙 1 9 を剥がして迅速に手帳やノートに貼り付けることができるという利点がある。また、タックシート 1 3 の幅は、手帳やノートの幅に合わせて例えば 7 0 mm 程度であってよい。

【0039】

次に、本実施の形態のテキストデータ処理装置であるパーソナルコンピュータ 1 1 0 およびカッティングプリンタ 1 1 からなるラベル製造装置 1 0 0 の制御系について、図 1 を参照して説明する。

【0040】

パーソナルコンピュータ 1 1 0 は、本体部 1 3 0 およびディスプレイ 1 3 2 のほか図示しないキーボード、マウスなどの機器から構成されている。本体部 1 3 0 は、CPU 1 3 4 と RAM 1 3 6 とハードディスク (HD) 1 3 8 とを有しており、これらはバスによって相互に接続されているとともに、入出力インターフェイス 1 4 0 とも接続されている。また、本体部 1 3 0 は、本実施の形態に係る、例えば FD や CD-ROM である記録媒体に記録されたプログラムを読み込むための駆動装置 (図示せず) を有している。また、本体部 1 3 0 には通信回線 1 3 9 と接続するためにモデムなどの機器 (図示せず) が内蔵されている。

【0041】

CPU 1 3 4 はハードディスク 1 3 8 や RAM 1 3 6 から読み込んだプログラム、データのほかカッティングプリンタ 1 1 側から供給されたデータに基づいて所定の演算を行なう。RAM 1 3 6 は、テキストデータや画像データのほかに、CPU 1 3 4 による演算結果などを一時的に記憶する。

【0042】

ハードディスク 138 には、ファイル管理ツールを含む汎用的な OS および汎用的なブラウザソフトウェアがインストールされている。ファイル管理ソフトウェアによって、パーソナルコンピュータ 110 内のファイルに対して、削除、コピー、移動、名前の変更などのファイル処理を行うことができる。

【0043】

ブラウザソフトウェアは、ハードディスク 138、FD および CD-ROM などに記憶されたテキストデータおよび／または画像データのほか、インターネット上の WWW サーバから通信回線 139 を介してパーソナルコンピュータ 110 に送られたテキストデータおよび／または画像データをディスプレイ 132 に表示することが可能である。ここで、テキストデータは、一例として HTML (hypertext markup language) によって表示された文字列データであり、また、画像データは、HTML 文書内にタグで貼り付けられた GIF、JPEG、BMP などの形式のデータである。

【0044】

図 6 に、ディスプレイ 132 へのブラウザソフトウェアの表示例を示す。ブラウザのデータ表示領域 150 には HTML 文書が表示されており、その左上部および左下部には GIF 画像 152、153 が貼り付けられている。また、その他の領域にはテキスト 154 が表示されている。

【0045】

また、ハードディスク 138 には、上述したソフトウェアのほかに、汎用的なワープロソフトウェア、描画ソフトウェアなどのテキストデータおよび／または画像データを処理できるソフトウェアがインストールされていてよい。例えばワープロソフトウェアは、当該ソフトウェア独自の形式で記憶されたテキストデータを編集することが可能であるとともに、GIF、JPEG、BMP などの画像データをテキストデータとともに表示することができるものであってよい。

【0046】

さらに、ハードディスク 138 には、カッティングプリンタ 11 でラベルを製造するためのエディタソフトウェア（エディタ）がインストールされている。エ

ディタは、これに取り込まれたテキストデータや画像データを所望のレイアウトで編集した後、タックシート 13 に印刷しこれを所望の位置でカットするためのソフトウェアであり、ユーザはパーソナルコンピュータ 110 を操作してシートに印刷する画像内容やカット位置などをディスプレイ 132 を見ながら編集することができる。

【0047】

エディタにテキストデータや画像データを取り込むには、使用する OS やアプリケーションに応じて様々な方法が考えられる。例えば、ブラウザのデータ表示領域 150 内の所定矩形範囲をマウスによって指定する、クリップボードを用いてコピー&ペーストを行う、ファイル管理ツールを用いて所望のデータを（例えばドラッグ&ドロップにより）選択するなどの様々な方法がある。なお、ディスプレイ 132 に表示された画像データがエディタに取り込まれる場合には、画像データはディスプレイ 132 と同じ画面解像度をもったデータとしてエディタに送られる。

【0048】

また、エディタは、これに取り込まれたテキストデータや画像データをその種類および組合せに応じて自動的に編集する機能を有している。詳しくは後述するが、エディタは、自動編集機能として、取り込まれた個々のデータがテキストデータおよび画像データのいずれであるのかを識別する機能、識別されたデータがテキストデータのみ、画像データのみ、テキストデータと画像データとの両方のいずれであるかを判定する機能、および、その判定結果に応じてそれぞれ異なるパターンにしたがってデータを配置する機能を有している。

【0049】

エディタは、データ配置機能の一部として、例えば、テキストデータだけが取り込まれた場合には、複数のテキストデータの区別が付くように各データの間に改行コードを 1 つ挿入して、ディスプレイ 132 に表示された所定領域内での相対位置に応じたアドレスに個々のテキストデータを配置する。また、その際に各テキストデータを整形して、表示されたテキストが見やすくなるようにする。特に HTML 文書から取り込んだテキストデータは、各表示行ごとに改行コードが

挿入されていたり、インデントが多用されていることが多く、これを異なる行内表示文字数で印刷すると行の途中で文章が改行されたり、スペース文字によって行頭位置が不揃いになったりして非常に読みづらい文章となってしまう。しかし、すべての改行コードやスペース文字を一律に削除してしまうと、かえって文章が読みづらくなる。そこで、本実施の形態では、所定の法則に従って、一定の条件を満足する改行コードおよびスペース文字だけを削除するようにしている。

【0050】

また、エディタは、データ配置機能の一部として、例えば、画像データだけが取り込まれた場合には、画像データが所定領域内に収まるようにこれを縮小或いは拡大してその所定領域内での相対位置に応じたアドレスに配置する。画像データを縮小する場合には、配置する際にドットを間引くなどの処理を行ってもよい。また、画像データを拡大する場合には、配置する際にドットを補間するなどの処理を行ってもよい。また、本実施の形態では、画像データが複数ある場合に、各画像データの配置順を決めて優先度が高いものから大きく表示されるように配置して、優先度が高い画像が見やすくなるようにしている。なお、複数の画像データ間に軽重をつけず、それぞれを独立した分割領域内において個別に拡大或いは縮小して配置するようにしてもよい。

【0051】

また、エディタは、データ配置機能の一部として、例えば、テキストデータと画像データの両方が取り込まれた場合には、所定領域を、テキストデータが配置されるべきテキスト領域と画像データが配置されるべき画像領域とに所定の占有比率で分割し、テキスト領域および画像領域それぞれにおいて上述したようにしてテキストデータおよび画像データをその相対位置に応じたアドレスに配置する。また、この占有比率を可変としておくことで、画像データとテキストデータのサイズの比率に応じた最適な占有比率を適宜選択してデータを見やすく表示することができる。

【0052】

図7に、テキストデータと画像データの両方が取り込まれた場合のディスプレイ132へのエディタの表示例を示す。エディタのデータ表示領域156内には

タックシート 1 3 の幅と等しい幅（ディスプレイ 1 3 2 上では拡大或いは縮小されて表示されている）を有する印刷領域 1 5 7 が表示されており、印刷領域 1 5 7 は余白部分を除いて画像領域 1 5 8 とテキスト領域 1 5 9 とに分割されている。画像領域 1 5 8 内には所定のパターンにしたがって配置された 1 または複数の画像データが表示され、テキスト領域 1 5 9 内には所定のパターンにしたがって配置された 1 または複数のテキストデータが、ユーザが設定した書体、文字サイズ、文字間隔、行間隔で表示される。なお、HTML 文書などでテキストデータ中の文字の書体や色、文字サイズが指定されている場合には、それをそのまま取り込んで表示してもよい。

【0 0 5 3】

印刷領域 1 5 7 はタックシート 1 3 を所定の拡大縮小率で拡大或いは縮小した状態でディスプレイ 1 3 2 上に表示されており、ユーザはその拡大縮小率を任意に変更することが可能である。また、画像領域 1 5 8 およびテキスト領域 1 5 9 についても、印刷領域 1 5 7 と同じ拡大縮小率で拡大或いは縮小されている。印刷領域 1 5 7 のディスプレイ 1 3 2 上での範囲は、印刷領域 1 5 7 の左上角部を原点としたアドレスとして RAM 1 3 6 の所定領域に印刷データの一部として記憶されている。また、画像領域 1 5 8 およびテキスト領域 1 5 9 内に配置された画像データおよびテキストデータの各アドレスについても、RAM 1 3 6 の所定領域に印刷データの一部として記憶されている。

【0 0 5 4】

印刷領域 1 5 7 は、長さ 1 0 0 mm 程度ごとに自動的に改ページ処理される。つまり、改ページ処理を行わないと、印刷領域 1 5 7 内に表示されるテキストデータのサイズが大きい場合などには、タックシート 1 3 が切断されることなく非常に長い領域にわたって印刷が行われ、これを手帳などに貼り付けるのに不便となる。そこで、長さ 1 0 0 mm 程度ごとに自動的に改ページ処理を行って、次のページの印刷領域 1 5 7 にデータが配置されるようにするとともに、タックシート 1 3 に 1 0 0 mm 程度印刷が行われるごとにタックシート 1 3 をカットするようにしている。なお、改ページ位置は、ユーザが任意の場所を選択することも可能である。改ページ位置のアドレスについても、RAM 1 3 6 の所定領域に印刷

データの一部として記憶されている。

【0055】

また、エディタは、改ページ位置だけではなく、テキストおよび画像の印刷が終了した際にもタックシート13をカットするような命令を出力する。これにより、1ページに満たない小さなサイズのデータを印刷する場合や、改ページされた次のページのわずかな領域だけにデータが印刷される場合などでも印刷された領域の下に余白が生じることがなくなるので、タックシート13が無駄に使用されるのを防止することができる。また、カッティングプリンタ11から排出されたタックシート13をユーザが切断するという手間が不要になる。このタックシートの切断位置アドレスについても、RAM136の所定領域に印刷データの一部として記憶されている。

【0056】

エディタのデータ表示領域156の上方には付属情報表示領域160が設けられており、データを取り込んだ日付やデータソースのURLが表示されるようになっている。また、コマンド表示領域161には、「印刷」、「保存」、「文字」、「レイアウト」の各コマンドメニューが表示されており、ユーザはこれらから所望のコマンドを選択して実行することができる。ユーザはデータ表示領域156に表示されたテキストおよび画像のレイアウトを確認し、必要であればレイアウトを変更したり文字列を入力或いは削除するなどの編集作業を行うことができる。このとき、印刷領域157にはテキストおよび画像がWYSWYG形式で表示されるので、編集作業を簡易に行うことができる。そして、印刷コマンドを選択することによりエディタで編集された印刷データ（画像データ、テキストデータおよびこれらのアドレスデータのほかに、上述した印刷範囲データ、改ページ位置データ、シート切断位置データなど）がカッティングプリンタ11のRAM118に送られて、カッティングプリンタ11のCPU114からの命令によりタックシート13の印刷および切断処理が行われる。

【0057】

カッティングプリンタ11は、その入出力インターフェイス112がパーソナルコンピュータ110の入出力インターフェイス140と接続されている。さら

に、入出力インターフェイス 112 には、CPU 114、ROM 116、RAM 118 のほか、サーマルヘッド 44 (図 2 および図 3 参照) を駆動するためのヘッド駆動回路 120、第 1 駆動モータ 21 および第 2 駆動モータ 35 (ともに図 3 参照) をそれぞれ駆動するためのモータ駆動回路 122、124、および、ソレノイド 41 (図 4 参照) を駆動するためのソレノイド駆動回路 126 などが接続されている。

【0058】

ROM 116 には、カッティングプリンタ 11 の動作を制御するためのプログラムのほか必要なデータが記憶されている。CPU 114 は ROM 116 から読み込んだプログラム、データのほかパーソナルコンピュータ 110 側から供給されたデータに基づいて所定の演算を行い、また、ヘッド駆動回路 120 などに制御信号を供給する。RAM 118 は、パーソナルコンピュータ 110 側から供給されたデータや CPU 114 による演算結果などを一時的に記憶する。

【0059】

本実施の形態では、パーソナルコンピュータ 110 内の CPU 134 が、改行コード検出手段、所定コード検出手段、改行コード削除手段、スペース文字検出手段、スペース数判定手段、複数スペース文字削除手段、行頭スペース文字検出手段、行頭スペース文字削除手段、改行コード・スペース文字検出手段、空白行フラグ制御手段、および、全スペース文字削除手段に対応している。また、パーソナルコンピュータ 110 内の RAM 136 が、スペースカウンタ、および、空白行フラグ記憶手段に対応している。

【0060】

次に、上述のように構成された本実施の形態のテキストデータ処理装置であるコンピュータ 110 およびカッティングプリンタ 11 からなるラベル製造装置 100 を用いてラベルを製造する具体的な手順について、図 8～図 17 をさらに参照して説明する。図 8 は、テキストデータと画像データとの識別処理に関するフローチャートである。図 9 は、テキストデータの整形処理に関するフローチャートである。図 10 は、テキストデータの改ページおよびタックシート切断処理に関するフローチャートである。図 11～図 13 は、本実施の形態によってテキス

トデータがどのように整形されるかを説明するための図である。図 1 4 は、画像データの配置処理に関するフローチャートである。図 1 5 は、図 1 4 に基づく画像データの配置例を印刷イメージとして模式的に示す図である。図 1 6 は、テキストデータと画像データとが混在した場合のこれら配置処理に関するフローチャートである。図 1 7 は、図 1 6 に基づくデータの配置例を印刷イメージとして模式的に示す図である。

【 0 0 6 1 】

まず、テキストデータと画像データとの識別処理について、図 8 に基づいて説明する。図 8 では、まず、ステップ S 1 において、パーソナルコンピュータ 1 1 0 を操作して、ブラウザおよびワープロソフトの少なくともいずれか 1 つを起動し、テキストデータおよび／または画像データをディスプレイ 1 3 2 に表示させ、ブラウザまたはワープロソフトからのコピー＆ペーストなどの何らかの方法によって、ディスプレイ 1 3 2 に表示されたデータから所望のデータを選択する。なお、ブラウザなどを用いる代わりに、ファイル操作ツールでファイルをドラッグ＆ドロップするなどの方法で処理すべきデータを選択してもよい。そして、このようにデータまたはファイルが選択された状態で所定の起動操作を行うことによってエディタが起動する。

【 0 0 6 2 】

次に、ステップ S 2 において、選択されたデータにフォーマット識別子があるかどうか判断される。フォーマット識別子は、データの種類 (html、txt、gif、bmp など) を示すものであって、データの先頭部分に書き込まれたテキストデータである。フォーマット識別子は、ブラウザまたはワープロソフトからのコピー＆ペーストによってデータが選択された場合には、データが表示されていたブラウザなどのアプリケーションによって書き込まれる。もしフォーマット識別子があれば (S 2 : YES) ステップ S 5 に進み、フォーマット識別子がなければ (S 2 : NO) ステップ S 3 に進む。フォーマット識別子がない場合とは、例えば、ファイル操作ツールでドラッグ＆ドロップによりファイルが選択されてファイルの内容がエディタ側で直ぐには参照できない場合である。

【 0 0 6 3 】

ステップ S 3 においては、選択されたデータがファイルであるかどうか判断される。もしファイルであれば (S 3 : Y E S) ステップ S 4 に進み、ファイルでなければ (S 3 : N O) ステップ S 1 1 に進む。ファイルでない場合とは、選択されたものが例えば単なる U R L や単なるフォルダ名である場合である。ステップ S 4 においては、ファイルの拡張子 (ファイル名において、.txt 、 .html 、 .gif 、 .bmp などのピリオドの後に置かれておりファイルの種類を示すもの) に基づいてそのデータのフォーマットを判定することとされる。

【 0 0 6 4 】

そして、ステップ S 5 においては、データのフォーマットが H T M L 或いは R T F フォーマットであるかどうか判断される。具体的には、データの先頭部分にあるフォーマット識別子或いはファイル拡張子が C P U 1 3 4 で判別される。もし H T M L または R T F フォーマットであればステップ S 6 に進み (S 5 : Y E S) 、 H T M L または R T F フォーマットでなければステップ S 7 に進む。ステップ S 6 においては、H T M L または R T F 文書中の図形タグ以下のデータが画像データとして、それ以外のデータがテキストデータとして識別されて取り込まれる。これらデータの種類 (画像データであるかテキストデータであるか) を示すデータは、R A M 1 3 6 の所定領域に個別データ識別データとして、個々の画像データまたはテキストデータが記憶されたアドレスと関連づけられて記憶される。

【 0 0 6 5 】

ステップ S 7 においては、ステップ S 5 と同様の手順により、データのフォーマットが B M P 、 G I F 、 J P E G などの画像フォーマットであるかどうか判断される。もし画像フォーマットであればステップ S 8 に進み (S 7 : Y E S) 、画像フォーマットでなければステップ S 9 に進む。ステップ S 8 においては、そのデータが画像データであると識別されて取り込まれる。そして、ステップ S 6 の場合と同様に、当該データが画像データであることを示すデータが R A M 1 3 6 の所定領域に個別データ識別データとして、個々の画像データが記憶されたアドレスと関連づけられて記憶される。

【 0 0 6 6 】

ステップS9においては、ステップS5と同様の手順により、データのフォーマットがテキストフォーマットであるかどうか判断される。もしテキストフォーマットであればステップS10に進み（S9：YES）、テキストフォーマットでなければステップS11に進む。ステップS10においては、そのデータがテキストデータであると識別されて取り込まれる。そして、ステップS6の場合と同様に、当該データがテキストデータであることを示すデータがRAM136の所定領域に個別データ識別データとして、個々のテキストデータが記憶されたアドレスと関連づけられて記憶される。

【0067】

次に、ステップS11においては、HTML或いはRTFフォーマット、画像フォーマットおよびテキストフォーマットのいずれのフォーマットでもないデータや、フォーマット識別子がなくファイルでもないデータが無効フォーマットと判断され、選択されたデータの取り込みが中止される。その際、無効フォーマットである旨或いはデータの取り込みを中止した旨をディスプレイ132に表示するなどの報知動作を行う。

【0068】

上述したようなステップをエディタに取り込まれた個々のデータについて繰り返して行うことにより、個々のデータがテキストデータ或いは画像データのいずれであるかを識別することが可能である。なお、ワープロソフトによって形成されるデータ（これを例えば「ABCフォーマット」と称することにする）をもエディタで扱う場合には、ステップS5において、そのデータのフォーマットがABC、HTML或いはRTFフォーマットであるかを判断すればよい。

【0069】

次に、RAM136に記憶された個別データ識別データをCPU134に読み込むことによって、エディタに取り込まれたデータが、テキストデータのみ、画像データのみ、および、テキストデータと画像データとの両方のいずれであるかを判断することが可能である。そして、その判断の結果は、RAM136の所定領域に組合せデータ判定データとして記憶される。なお、図8のステップS5でHTML或いはRTFフォーマットであると判断された場合であって、その文書

中に図形タグが含まれている場合にはテキストデータと画像データとの両方が存在すると判定してもよい。

【0070】

次に、上述した処理によってエディタに取り込まれたデータがテキストデータのみであると判断された場合の、エディタによるテキストデータの処理について説明する。組合せデータ判定データがエディタに取り込まれたデータがテキストデータのみであることを示している場合、図7に示す印刷領域157の余白部分を除く領域に対応するアドレス領域にテキストが配置されることになる。まず1番目のテキストデータに対して以下に説明する整形処理が施され、それが印刷領域157の上位アドレスに配置されてそのアドレスがRAM136の所定領域に書き込まれ、印刷領域157に1番目のテキストデータが表示される。そして、改行コードが挿入されて1行空けた後に、2番目のテキストデータに整形処理が施され、1番目のテキストデータの続きのアドレスに配置されてそのアドレスがRAM136の所定領域に書き込まれ、印刷領域157に2番目のテキストデータが表示される。以下、最後のテキストデータまで同様の処理が繰り返される。この間、印刷領域157の長さが所定長さを超えるごとにタックシート13が切断され、最後のテキストデータの印刷が終了した際にもその直後でタックシート13が切断されるように、改ページ位置データ、シート切断位置データが書き込まれるような制御が行われる。以下、図9～図13を参照して詳述する。

【0071】

まず、テキストデータの整形処理について、図9を参照して説明する。まず、ステップT1において、RAM136の所定領域に記憶されるスペースカウンタがゼロに初期化され、続いて、ステップT2において、RAM136の所定領域に記憶される空白行フラグがオンにされ、さらに、ステップT3において、RAM136の所定領域に記憶される文字ポインタがテキストデータの先頭の文字コードに移される。ここで、スペースカウンタは、テキストデータ中にいくつのスペース文字（全角スペース、半角スペース、タブコードなど）が連続して存在しているかを表すカウンタである。また、空白行フラグは、1つの論理行にスペース文字以外の文字が存在するかどうかを表すフラグである。

【0072】

次に、ステップT4において、1つのテキストデータの整形処理が終了したか、つまり文字ポインタ位置がテキストデータの最終文字位置を超えているかどうか判断される。もし超えていれば（T4：YES）当該テキストデータの処理を終了し、超えていなければ（T4：NO）ステップT5に進む。

【0073】

ステップT5においては、文字ポインタが存在する文字コードが改行コードであるかが判断される。もし改行コードであれば（T5：YES）ステップT6に進み、改行コードでなければ（T5：NO）ステップT11に進む。

【0074】

ステップT6においては、当該改行コードの直前にある文字が句読点などの区切り記号（例えば、「。」、「、」、「、」、「。」、「：」、「；」など）であるか、或いは、当該改行コードの直後（1～3文字程度のスペース文字を挟んだ直後であってもよい）にある文字が簡条書きの各行の先頭に通常使用される簡条書き記号（例えば、「・」、「1.」、「(1)」、「①」、「A.」など）であるかが判断される。もしこの条件に当てはまる場合には（T6：YES）ステップT8に進み、もしこの条件に当てはまらない場合には（T6：NO）ステップT7に進み、当該改行コードは削除される。

【0075】

ステップT6で直前に区切り記号のある改行コードを削除しないようにしたのは、これを削除すると、段落の区切りが分かりづらくなり、文章が途切れることなく一続きに非常に長くつながってしまい、読みづらくなるからである。また、直後に簡条書き記号がある改行コードを削除しないようにしたのは、簡条書きされた文章を改行することなく同じ行内に表示するのは非常に読みづらくなるからである。

【0076】

次に、ステップT8において、空白行フラグがオンであるかどうか判断される。もし空白行フラグがオンであれば（T8：YES）ステップT9に進み、空白行フラグがオフであれば（T8：NO）ステップT10に進む。ステップT8

で空白行フラグがオンであるときには、その論理行にスペース文字以外の文字が存在していない（言い換えると、当該論理行の改行コードの前に存在するのはすべてスペース文字である）から、ステップT 9において、その論理行のすべてのスペース文字（直前の改行コードの直後または文頭から現在位置までのスペース文字）が削除される。その後、ステップT 1 0に進んで空白行フラグがオンにされ、ステップT 1 8に進む。

【0 0 7 7】

また、ステップT 1 1においては、文字ポインタが存在する文字コードがスペース文字であるかが判断される。もしスペース文字であれば（T 1 1 : Y E S）ステップT 1 2に進み、スペース文字でなければ（T 1 1 : N O）ステップT 1 3に進む。ステップT 1 2においては、スペースカウンタが1だけ加算され、その後ステップT 1 9に進む。

【0 0 7 8】

ステップT 1 3においては、スペースカウンタのカウント値が2以上となっているかどうか判断される。ここで、カウント値2以上としたのは、1つのスペース文字をすべて削除してしまうと、欧文など単語間に半角スペースが挿入されている文書からスペースを除くと単語の区別がつかなくなりテキストの判読ができなくなるおそれがあるからである。そして、スペースカウンタのカウント値が2以上であれば（T 1 3 : Y E S）ステップT 1 4に進み、スペースカウンタのカウント値が2未満であれば（T 1 3 : N O）ステップT 1 5に進む。

【0 0 7 9】

ステップT 1 4においては、スペースカウンタのカウント値が2以上であるので、元のテキストデータ中のインデントなどによって必要以上に多くのスペース文字が挿入されていると判断してそのスペース文字を削除し、ステップT 1 7に進む。

【0 0 8 0】

また、ステップT 1 5においては、空白行フラグがオンであるかどうかを判断し、もし空白行フラグがオンであれば（T 1 5 : Y E S）ステップT 1 6に進み、空白行フラグがオフであれば（T 1 5 : N O）ステップT 1 0に進む。ステッ

ブ T 1 5 で空白行フラグがオンである場合とは、後のステップ T 1 7 で空白行フラグがオフにされることから、文字ポインタが行頭にあるときか或いは文字ポインタが行頭から 2 文字目にあって 1 文字目がスペース文字であった場合である。そこで、ステップ T 1 6 において、文字ポインタの直前にあるスペース文字を削除し、欧文の場合など単語間などに故意に挿入されたスペース文字が削除されないようにしつつ、行頭にあるスペース文字を削除している。これにより、行頭がすべての行において揃うことになるので、行頭に統一感が生み出され、テキストデータが見やすくなる。

【 0 0 8 1 】

次に、ステップ T 1 7 において空白行フラグがオフにされた後、ステップ T 1 8 においてスペースカウンタがゼロに初期化される。さらに、ステップ T 1 9 において、文字ポインタが次の文字コードに移されて、ステップ T 4 に戻って同様のステップが繰り返される。

【 0 0 8 2 】

本実施の形態によってテキストデータがどのように整形されるかの一例を図 1 1 ～図 1 3 に基づいて説明する。なお、これらの図において、下向きの矢印（↓）は改行コードを表すものであり、実際に印刷されるものではない。

【 0 0 8 3 】

図 1 1 は、ブラウザに示された元のテキストデータであり、各行末には改行コードが付加されている。また、3 ～ 6 行目はインデント処理されており、7 行目には改行コードのみが存在している。このテキストデータを単に図 1 1 よりも少ない行内文字数で表したものが図 1 2 である。図 1 2 に示されたテキストデータは、行の途中で改行されたり、不要な改行がなされていたり、インデントされた行にスペースが残ってしまって非常に見づらい。

【 0 0 8 4 】

本実施の形態にしたがって処理したテキストデータを示したのが図 1 3 である。図 1 3 から明らかなように、1 行目先頭のスペースは、ステップ T 4 ～ステップ T 1 1、ステップ T 1 2 を経てステップ T 4 に戻った後、ステップ T 1 3、ステップ T 1 5、ステップ T 1 6 を経て削除される。また、インデント処理された

3～6行目の各行先頭のスペースは、ステップT4～ステップT11、ステップT12、ステップT19を経てステップT4に戻る処理をスペースの数だけ繰り返した後、ステップT11、ステップT13、ステップT14を経て削除される。また、3～5行目の行末の改行コードと7行目の改行コードは、ステップT4～ステップT7を経て削除される。一方、2行目、6行目、8行目～11行目の行末の改行コードは、ステップT4～ステップT6を経てステップT7が実行されないで削除されずに残る。このように、本実施の形態によると、行内文字数が図11よりも少なくなったとしても、所定の場合以外は行の途中で改行されることがなく、インデントされていた行にスペースが残ることもない。また、「・」記号が行頭にあって箇条書きされた行は、改行コードが削除されることなくそのまま箇条書きの形式で残され、句点「。」の直後にある改行コードは削除されないで、各行の区別が分かりやすい。また、行頭にあった1文字分のスペースも削除されるので、すべての行の行頭に統一感が生み出されている。また、図11において7行目にあった改行コードも削除されて、テキストがコンパクトになっている。つまり、本実施の形態によると、煩雑な作業を必要とせずに、テキストデータを、その行内表示文字数が変わった場合であっても見やすくコンパクトに表示することができる。そのため、タックシート13のような幅の狭い領域に印刷されたテキストが非常に読みやすくなる。

【0085】

次に、テキストデータのみが配置される場合のタックシート13の改ページおよびシート切断処理について図10を参照して説明する。なお、ここでは、1つのテキストデータのサイズが大きく1つのテキストデータが複数のページにまたがって印刷される場合について説明するが、複数のテキストデータが印刷される場合も各データの間には改行コードのみの行が1つ設けられる点以外は同様に処理される。

【0086】

まず、ステップU1において、オブジェクト先頭のY座標（1ページ目のタックシート13の長さ方向についての印刷開始位置の座標）が変数Yに格納され、続いて、ステップU2において、RAM136の所定領域に記憶された行ポイン

タが先頭行に移される。

【 0 0 8 7 】

次に、ステップU 3 において、未処理の行が存在するか、つまり行ポインタ位置がテキストデータの最終行位置を超えているかどうか判断される。もし超えていれば（U 3 : N O）ステップU 5 に進み、現在位置でタックシート 1 3 が切断されるようにその切断位置が R A M 1 3 6 の所定領域に書き込まれる。もし超えていなければ（U 3 : Y E S）ステップU 4 に進む。

【 0 0 8 8 】

ステップU 4 においては、変数 Y に現在行の行幅を足した値が、そのページ末の Y 座標（タックシート 1 3 の各ページの長さ方向の印刷終了位置）を超えているかどうか判断される。もし、超えていなければ（U 4 : N O）ステップU 9 に進む。一方、もし超えていれば（U 4 : Y E S）ステップU 6 に進み、現在位置でタックシート 1 3 が切断されるようにその切断位置が R A M 1 3 6 の所定領域に書き込まれる。そして、ステップU 7 において、次のページの先頭（印刷開始位置）の Y 座標が変数 Y に格納された後、ステップU 8 において、タックシート 1 3 が搬送されて印刷開始位置の頭出しがされるようなデータが R A M 1 3 6 の所定領域に書き込まれてからステップU 9 に進む。

【 0 0 8 9 】

ステップU 9 においては、行ポインタが存在する現在行のテキストデータがタックシート 1 3 に印刷されるように、その行のテキストデータが R A M 1 3 6 の所定領域に書き込まれる。そして、ステップU 1 0 に進んで、次の行の印刷のために、タックシート 1 3 が現在行の行幅分だけ前方に搬送されるようなデータが R A M 1 3 6 の所定領域に書き込まれる。

【 0 0 9 0 】

次に、ステップU 1 1 において、変数 Y に現在行の行幅を足した値が新たな変数 Y として格納され、ステップU 1 2 において、行ポインタが次の行に移された後、ステップU 3 に戻る。

【 0 0 9 1 】

このような処理により、テキストデータが複数ページにまたがる場合であって

も、タックシート 1 3 を所定の長さごとに切断して手帳などに貼り付けやすいラベルを得ることができる。また、テキストデータが 1 ページに満たない長さで終了した場合には、その場所でタックシート 1 3 を切断するようにしたので、切断されたタックシート 1 3 に余白部分が生じることがなくコンパクトな印刷済みラベルを低コストで得ることができる。また、カッティングプリンタ 1 1 から排出されたラベルの余白部分を切断するという手間が不要になる。

【0 0 9 2】

次に、図 8 に関連して説明した処理によってエディタに取り込まれたデータが画像データのみであると判断された場合の、エディタによる画像データの処理について説明する。組合せデータ判定データがエディタに取り込まれたデータが画像データのみであることを示している場合、図 7 に示す印刷領域 1 5 7 の余白部分を除く領域に対応するアドレス領域に画像データが配置されることになる。本実施の形態では、選択された複数の画像データが優先順位を付けられて、優先順位の高いものから順にタックシート 1 3 の 1 ページ内に対応したアドレス領域に配置されていく場合について説明する。以下、図 1 4 ～図 1 5 を参照して詳述する。

【0 0 9 3】

本実施の形態では、図 1 5 に示す、タックシート 1 3 への 1 ページ分の矩形印刷領域に対応したアドレス領域に、選択された n 個の矩形画像データを配置するものとする。これら n 個の画像データの配置の優先順位は、本実施の形態ではそのサイズ（バイト数）の大きい順に決められる。つまり、CPU 1 3 4 が選択された画像データの大きさをそれぞれ比較することによって優先順位を決定し、その順番が RAM 1 3 6 の所定領域に書き込まれ、RAM 1 3 6 に書き込まれた順番にしたがって最もサイズの大きい画像データから配置されていく。これによって、サイズの大きい画像データを大きく表示および印刷させることが可能となる。なお、画像データの配置の優先順位は、画像データが選択された順であってもよいし、ユーザが任意に決めてもよく、いずれの方法によっても画像データの配置の優先順位は CPU 1 3 4 によって決定され、RAM 1 3 6 の所定領域に書き込まれる。

【0094】

そして、RAM 136に書き込まれた順番にしたがって、印刷領域の画像未配置領域に対応するアドレス領域内に元の画像データを相似拡大または相似縮小した相似画像データが配置される。このとき、各画像データは画像未配置領域において最大のサイズとなるように相似拡大または相似縮小される。そして、その配置方向はすべての画像データについて共通にされているので、観察方向を見る画像によって変える必要なく1枚のタックシートに印刷された複数の画像を一目で把握できるようになる。

【0095】

本実施の形態では、図15に示すように、タックシートの印刷領域の幅に対応した長さ（ドット数）をX、タックシート13の1ページ分の長さに対応した長さをYとし、この（X，Y）の印刷領域に対応するアドレス領域内にn個の画像データがそれぞれ相似拡大または相似縮小されて配置される。図の上側が画像の上側に対応している。ここで、n番目の画像データのブラウザに表示されていた元の大きさを（ x_n ， y_n ）とし、このデータが相似拡大または相似縮小された相似画像データのエディタ画面での大きさを（ $x_{n'}$ ， $y_{n'}$ ）とする。また、1または複数の相似画像データが配置されているときに、印刷領域の左上頂点を基準点として、幅Xのすべてが相似画像データで占められている位置までのY方向の距離をBとし、幅Yのすべてが相似画像データで占められている位置までのX方向の距離をAとしている。

【0096】

まず、図14に示すステップE1において、パラメータnが1とされるとともに、上述した変数AおよびBがゼロとされる。そして、ステップE2において、 $X - A$ の値が、n番目の画像データの幅 x_n の拡大或いは縮小後の幅 $x_{n'}$ として格納される。

【0097】

次に、ステップE3において、ステップE2で決められた拡大或いは縮小後の画像データの幅 $x_{n'}$ に基づいて、この画像データを相似拡大或いは相似縮小したときの長さ $y_{n'}$ を求め（ $y_{n'} = (y_n / x_n) * x_{n'}$ ）、長さ $y_{n'}$ と

$Y - B$ の値とを比較する。その結果、 $y_n' \leq Y - B$ であれば (E 3 : YES) ステップ E 4 に進み、 $y_n' > Y - B$ であれば (E 3 : NO) ステップ E 5 に進む。

【0098】

$y_n' \leq Y - B$ であるとは、例えば 1 番目の画像データについてはその拡大或いは縮小後の幅 x_1' を印刷領域の全幅 X としたときに、その拡大或いは縮小後の長さ y_1' が印刷領域の 1 ページの長さ Y 内に収まることを意味する。そこで、この場合には、ステップ E 2 およびステップ E 3 で定めた相似画像データ (x_n' , y_n') を画像未配置領域に配置することとし、ステップ E 4 において、 $B + y_n'$ を新たに B の値として格納する。

【0099】

また、 $y_n' > Y - B$ であるとは、例えば 1 番目の画像データについてはその拡大或いは縮小後の幅 x_1' を印刷領域の全幅 X としたときに、その拡大或いは縮小後の長さ y_1' が印刷領域の 1 ページの長さ Y 内に収まらないことを意味する。そこで、この場合には、 X 方向を相似変形の基準とするのをやめて、 Y 方向を相似変形の基準とするようにする。そこで、ステップ E 5 において、 $Y - B$ の値が、拡大或いは縮小後の画像データの長さ y_n' として格納される。

【0100】

次に、ステップ E 6 において、ステップ E 5 で決められた拡大或いは縮小後の画像データの長さに基づいて、この画像データを相似拡大或いは相似縮小したときの幅 x_n' を求める ($x_n' = (x_n / y_n) * y_n'$)。そして、ステップ E 7 において、 $A + x_n'$ を新たに A の値として格納する。

【0101】

次に、ステップ E 8 において、ステップ E 2 およびステップ E 3 で定めた相似画像データ (x_n' , y_n')、または、ステップ E 5 およびステップ E 6 で定めた相似画像データ (x_n' , y_n') が画像未配置領域に対応するアドレス領域に (左上詰めで) 上位アドレスから配置される。このとき、元の画像データからの拡大縮小率とともに相似画像データの配置位置範囲を示すアドレスが元の画像データのアドレスと関連づけられて RAM 136 の所定領域に印刷データの一

部として書き込まれる。

【0 1 0 2】

次に、ステップE 9において、変数AがXと等しくなったか（ $A = X$ ）、或いは、変数BがYと等しくなったか（ $B = Y$ ）どうか判断される。もしこれらいずれかの条件が満たされていれば（E 9 : YES）1 ページ分の印刷領域が相似画像データで埋め尽くされたことになるので、ステップE 1 1に進んで、長さYでタックシート1 3が幅方向に切断されるようにその切断位置がRAM 1 3 6の所定領域に印刷データの一部として書き込まれる。

【0 1 0 3】

また、これらいずれの条件も満たされていなければ（E 9 : NO）、タックシート1 3の1 ページ分の印刷領域にまだ空白部分があることになるので、ステップE 1 0に進んで次の画像データがあるかどうか判断される。もし次の画像データがあれば（E 1 0 : YES）ステップE 1 2に進んでパラメータnを1だけ加算してステップE 2に戻る。もし次の画像データがなければ（E 1 0 : NO）ステップE 1 3に進む。

【0 1 0 4】

ステップE 1 3では、変数Aがゼロであるかどうか判断される。もし変数Aがゼロであれば（E 1 3 : YES）、1 ページ分の印刷領域の下方にはまったく印刷されない領域があることになるので、ステップE 1 4に進んで位置Bでタックシート1 3が幅方向に切断されるようにその切断位置がRAM 1 3 6の所定領域に印刷データの一部として書き込まれる。また、もし変数Aがゼロでなければ（E 1 3 : NO）、1 ページ分の印刷領域の長さYにわたって印刷が行われることになるので、ステップE 1 1に進んで高さYでタックシート1 3が幅方向に切断されるようにその切断位置がRAM 1 3 6の所定領域に印刷データの一部として書き込まれる。

【0 1 0 5】

このような手順によって、1または複数の画像データを、煩雑な編集作業を必要とすることなく印刷領域に対応するアドレス領域内に拡大或いは縮小して配置し、エディタへの見やすく且つコンパクトな表示を得ることが可能となる。

【0106】

次に、図8に関連して説明した処理によってエディタに取り込まれたデータがテキストデータと画像データの両方であると判断された場合の、エディタによるテキストデータおよび画像データの処理について説明する。組合せデータ判定データがエディタに取り込まれたデータがテキストデータと画像データの両方であることを示している場合、図7に示す印刷領域157の余白部分を除く画像領域158に対応するアドレス領域に画像データが配置され、テキスト領域159に対応するアドレス領域にテキストデータが配置されることになる。以下、図16～図17を参照して詳述する。

【0107】

本実施の形態では、図17に示す、タックシート13への1ページ分の矩形印刷領域を、テキストデータが配置されるテキスト領域と、画像データが配置される画像データとに所定の占有比率で分割し、それぞれの領域に対応するアドレス領域内でテキストデータおよび画像データを配置していく。分割のレイアウトは様々なものが考えられるが、本実施の形態では、占有比率に応じた位置でタックシート13の長さ方向に沿って印刷領域が2つの矩形領域に分けられる。テキストデータおよび画像データの各領域内での配置の手順は、上述した図9、図10、図14、図15で説明したものと同じでもよい。すなわち、テキスト領域では、複数のテキストデータが1つの改行コードを挟んで連続して配置され、各テキストデータに整形処理が施される。また、画像領域では、優先順位を付けられた複数の画像データがその順番に画像未配置領域で最大となるように相似拡大または相似縮小されて配置される。

【0108】

本実施の形態では、図17に示すように、タックシートの印刷領域の幅に対応した長さ（ドット数）を W 、タックシート13の1ページ分の長さに対応した長さを H とし、そのうち、画像領域の幅を X 、長さを Y とした（つまり、画像領域の占有比率は X/W であり、テキスト領域の占有比率は $(W-X)/W$ である）。そして、この (X, Y) の画像領域に対応するアドレス領域内に n 個の画像データがそれぞれ相似拡大または相似縮小されて配置され、 $(W-X, Y)$ のテキ

スト領域に対応するアドレス領域内に 1 または複数のテキストデータが配置される。ここで、 n 番目の画像データのブラウザに表示されていた元の大きさを (x_n, y_n) とし、このデータが相似拡大または相似縮小された相似画像データのエディタ画面での大きさを (x_n', y_n') とする。また、画像領域内に 1 または複数の相似画像データが配置されているときに、画像領域の左上頂点を基準点として、幅 X のすべてが相似画像データで占められている位置までの Y 方向のディスプレイ 1 3 2 上での距離を B とし、幅 Y のすべてが相似画像データで占められている位置までの X 方向のディスプレイ 1 3 2 上での距離を A としている。

【0 1 0 9】

まず、図 1 6 に示すステップ P_0 において、1 ページ分のタックシートの印刷領域を占有比率 R を用いてタックシート 1 3 の長さ方向に沿って分割し、画像領域の幅 X を求める ($X = (R / 100) * W$)。ステップ P_0 の後に行われるステップ $P_1 \sim$ ステップ P_8 は図 1 4 で説明したステップ $E_1 \sim$ ステップ E_8 と実質的に同じであるので、ここではその説明を省略する。

【0 1 1 0】

そして、ステップ P_9 において、変数 A が X と等しくなったか ($A = X$)、或いは、変数 B が Y と等しくなったか ($B = Y$) どうか判断される。もしこれらいずれかの条件が満たされていれば ($P_9 : YES$) 1 ページ分の画像領域が相似画像データで埋め尽くされたことになるので、ステップ P_{11} に進んで、テキスト領域にテキストデータを配置してステップ P_{12} に進む。テキストデータの配置手順は、上述した図 9 で説明したのと同じであるのでここではその説明を省略する。なお、ここでは、処理を簡単にするために、テキストデータがタックシート 1 ページ分のテキスト領域を超えて印刷されることがないと仮定している。

【0 1 1 1】

また、これらいずれの条件も満たされていなければ ($P_9 : NO$)、タックシート 1 3 の 1 ページ分の画像領域にまだ空白部分があることになるので、ステップ P_{10} に進んで次の画像データがあるかどうか判断される。もし次の画像データがあれば ($P_{10} : YES$) ステップ P_{13} に進んでパラメータ n を 1 だけ加算してステップ P_2 に戻る。もし次の画像データがなければ ($P_{10} : NO$)

ステップ P 1 4 に進んで、テキスト領域にテキストデータを配置する。このとき、画像領域内の相似画像データが配置されなかった空白領域にもテキストデータを配置することによって、空白領域を有効に利用してテキストデータをコンパクトに配置することができるようになる。

【 0 1 1 2 】

次に、ステップ P 1 5 では、変数 A がゼロであるかどうか判断される。もし変数 A がゼロでなければ (P 1 5 : N O)、タックシート 1 3 の 1 ページ分の長さによって画像が印刷されることになるので、ステップ P 1 2 に進んで高さ H、つまり 1 ページ分の長さでタックシート 1 3 が幅方向に切断されるようにその切断位置が R A M 1 3 6 の所定領域に印刷データの一部として書き込まれる。もし変数 A がゼロであれば (P 1 5 : Y E S)、1 ページ分の画像領域の下方にまったく画像が印刷されない領域があることになる。そこで、ステップ P 1 6 に進んで、変数 B がテキストデータの高さ (第 1 行位置から最下行位置までの距離) よりも大きいかどうか、つまり画像データがテキストデータよりも下方に突出しているかどうか判断される。もし変数 B がテキストデータの高さよりも大きければ (P 1 6 : Y E S) ステップ P 1 7 に進んで、位置 B でタックシート 1 3 が幅方向に切断されるようにその切断位置が R A M 1 3 6 の所定領域に印刷データの一部として書き込まれる。一方、変数 B がテキストデータの高さよりも大きくなければ (P 1 6 : N O) ステップ P 1 8 に進んで、テキストデータの高さでタックシート 1 3 が幅方向に切断されるようにその切断位置が R A M 1 3 6 の所定領域に印刷データの一部として書き込まれる。

【 0 1 1 3 】

このような手順によって、画像データとテキストデータとを印刷領域に対応するアドレス領域内に自動的に配置することができるので、煩雑な編集作業を必要とすることなく画像データおよびテキストデータを配置し、見やすく且つコンパクトな表示を得ることが可能となる。

【 0 1 1 4 】

このように、本実施の形態によると、ブラウザやワープロソフトなどの中のテキストデータや画像データを指定するという簡単な操作だけで、指定された画像

やテキストが見やすく表示および印刷されて、手帳やノートに貼るのに適したサイズのラベルを作成することができるようになる。従って、必要な情報だけを印刷して手帳などに貼り付けておくことにより、タックシートを無駄に使用することなく、必要な情報を手軽に持ち歩いていつでも参照することが可能となる。

【0115】

なお、上述の実施の形態では、本発明をタックシートに印刷することを例に説明したが、本発明は必ずしもこれに関する用途に限られるものではなく、テキストデータ処理一般に用いることが可能である。また、上述の実施の形態では、本発明のテキストデータ処理装置はパーソナルコンピュータによって構成されていたが、カッティングプリンタだけ或いカッティングプリンタとパーソナルコンピュータの両方で本発明のテキストデータ処理装置が構成されていてもよい。

【0116】

【発明の効果】

以上説明したように、請求項1、4、7によると、直前または直後に所定コードが存在しない改行コードだけを選択して自動的に削除することができるので、ユーザが個々のテキストデータから不要な改行コードだけを削除するという煩雑な作業を必要とせずに、テキストデータをその行内表示文字数が変わった場合であっても見やすく表示或いは印刷させることが可能となる。

【0117】

請求項2、5、8によると、2以上連続したスペース文字を検出してこれを自動的に削除することができるので、ユーザが個々のテキストデータから不要なスペース文字だけを削除するという煩雑な作業を必要とせずに、例えば単語間に故意に挿入された1つのスペース文字を削除することなく、テキストデータをその行内表示文字数が変わった場合であっても見やすく表示或いは印刷させることが可能となる。

【0118】

請求項3、6、9によると、行頭に1つだけ存在するスペース文字が削除されるので、単語間などに故意に挿入された1つのスペース文字が削除されないようにしつつ、行頭にあるスペース文字を削除することができる。そのため、行頭に

統一感が生み出され、テキストデータが見やすくなる。

【0 1 1 9】

請求項 1 0 によると、直前または直後に所定コードが存在しない改行コードだけを選択して自動的に削除することができるのと同時に、2 以上連続したスペース文字を検出して自動的に削除することができるので、ユーザが個々のテキストデータから不要な改行コードおよびスペース文字だけを削除するという煩雑な作業を必要とせずに、テキストデータをその行内表示文字数が変わった場合であっても見やすく表示或いは印刷させることが可能となる。

【0 1 2 0】

請求項 1 1 によると、空白行フラグを用いて論理行の行頭にある 1 つのスペース文字を削除することができるので、単語間などに故意に挿入された 1 つのスペース文字が削除されないようにしつつ、行頭にあるスペース文字を削除することができる。そのため、行頭に統一感が生み出され、テキストデータが見やすくなる。

【0 1 2 1】

請求項 1 2 によると、空白行フラグを用いて改行コード（または文頭）と改行コードとの間にスペース文字だけがある論理行のすべてのスペース文字を削除することができるので、効率的に不要なスペースを削除してテキストデータをその行内表示文字数が変わった場合であっても見やすく表示或いは印刷させることが可能となる。

【図面の簡単な説明】

【図 1】

本発明の一実施の形態のテキストデータ処理装置を含むラベル製造装置のブロック図である。

【図 2】

図 1 に示すカッティングプリンタの要部構成を示す平面図である。

【図 3】

図 1 に示すカッティングプリンタの要部断面図である。

【図 4】

図 1 に示すカッティングプリンタのカッティング機構部を示す要部側断面図である。

【図 5】

図 1 に示すカッティングプリンタにおいて、ロールシートユニットとカッティング機構部および画像形成機構部との概略的な位置関係を示す模式的な斜視図である。

【図 6】

図 1 に示すディスプレイへのブラウザの表示例を示す図である。

【図 7】

本発明の一実施の形態において、テキストデータと画像データの両方が取り込まれた場合のエディタの表示例を示す図である。

【図 8】

本発明の一実施の形態におけるテキストデータと画像データとの識別処理に関するフローチャートである。

【図 9】

本発明の一実施の形態におけるテキストデータの整形処理に関するフローチャートである。

【図 1 0】

本発明の一実施の形態におけるテキストデータの改ページおよびタックシート切断処理に関するフローチャートである。

【図 1 1】

本発明の一実施の形態によってテキストデータがどのように整形されるかを説明するための図である。

【図 1 2】

本発明の一実施の形態によってテキストデータがどのように整形されるかを説明するための図である。

【図 1 3】

本発明の一実施の形態によってテキストデータがどのように整形されるかを説明するための図である。

【図 1 4】

本発明の一実施の形態における画像データの配置処理に関するフローチャートである。

【図 1 5】

図 1 4 に基づく画像データの配置例を印刷イメージとして模式的に示す図である。

【図 1 6】

本発明の一実施の形態におけるテキストデータおよび画像データの配置処理に関するフローチャートである。

【図 1 7】

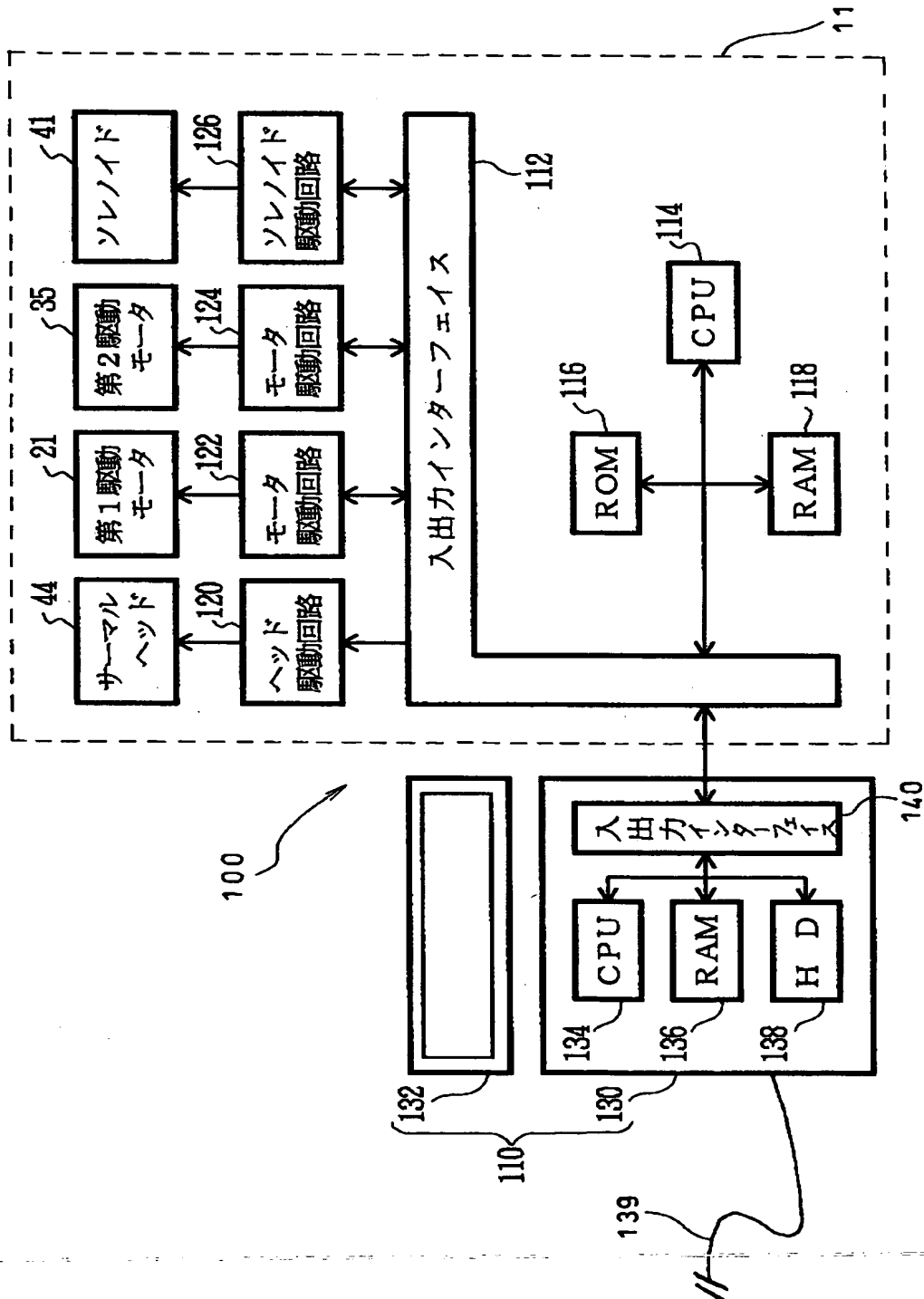
図 1 6 に基づくテキストデータおよび画像データの配置例を印刷イメージとして模式的に示す図である。

【符号の説明】

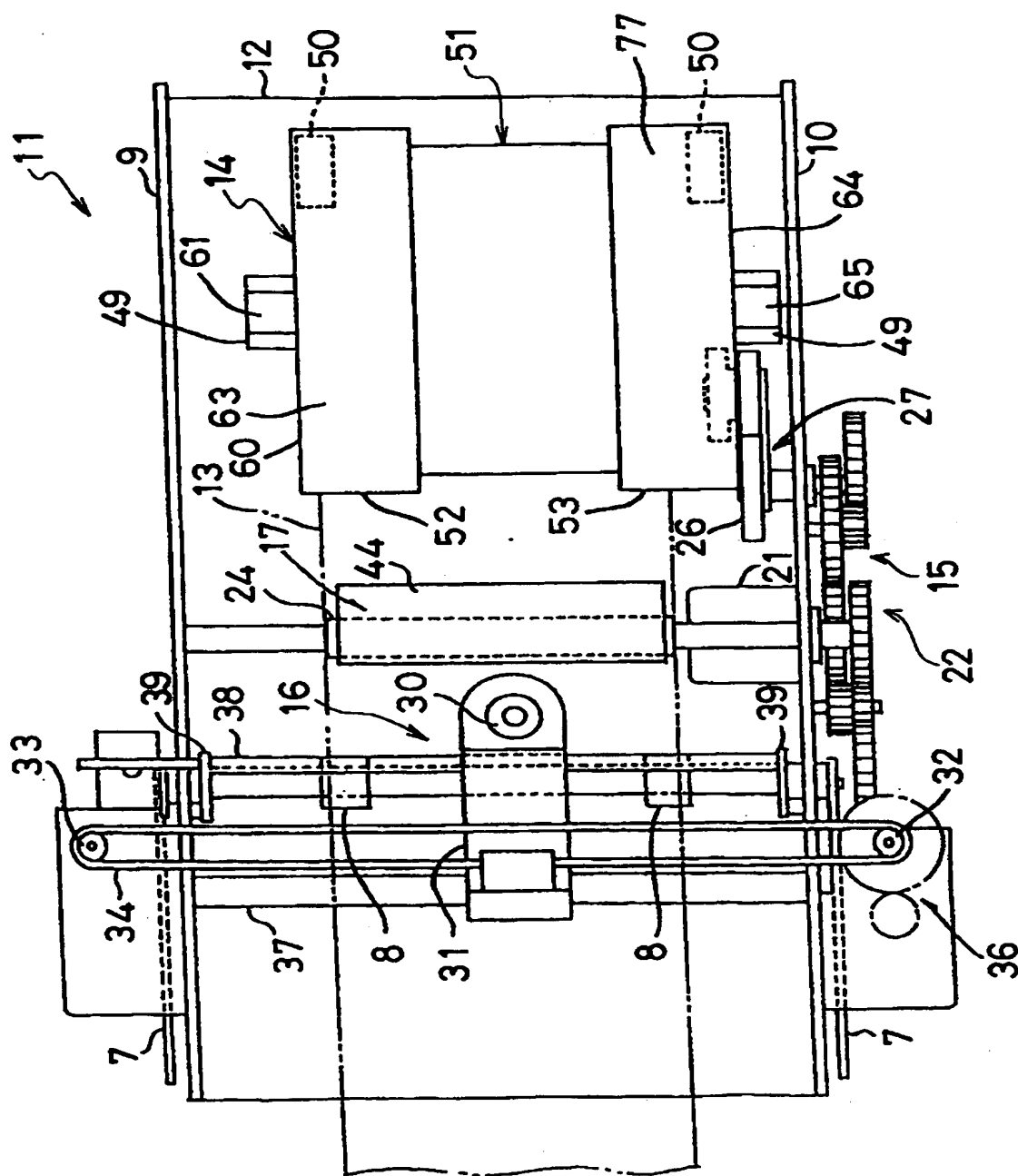
- 1 1 カッティングプリンタ
- 1 0 0 ラベル製造装置
- 1 1 0 パーソナルコンピュータ
- 1 3 2 ディスプレイ
- 1 3 4 CPU
- 1 3 6 RAM
- 1 5 7 印刷領域
- 1 5 8 画像領域
- 1 5 9 テキスト領域

【書類名】 図面

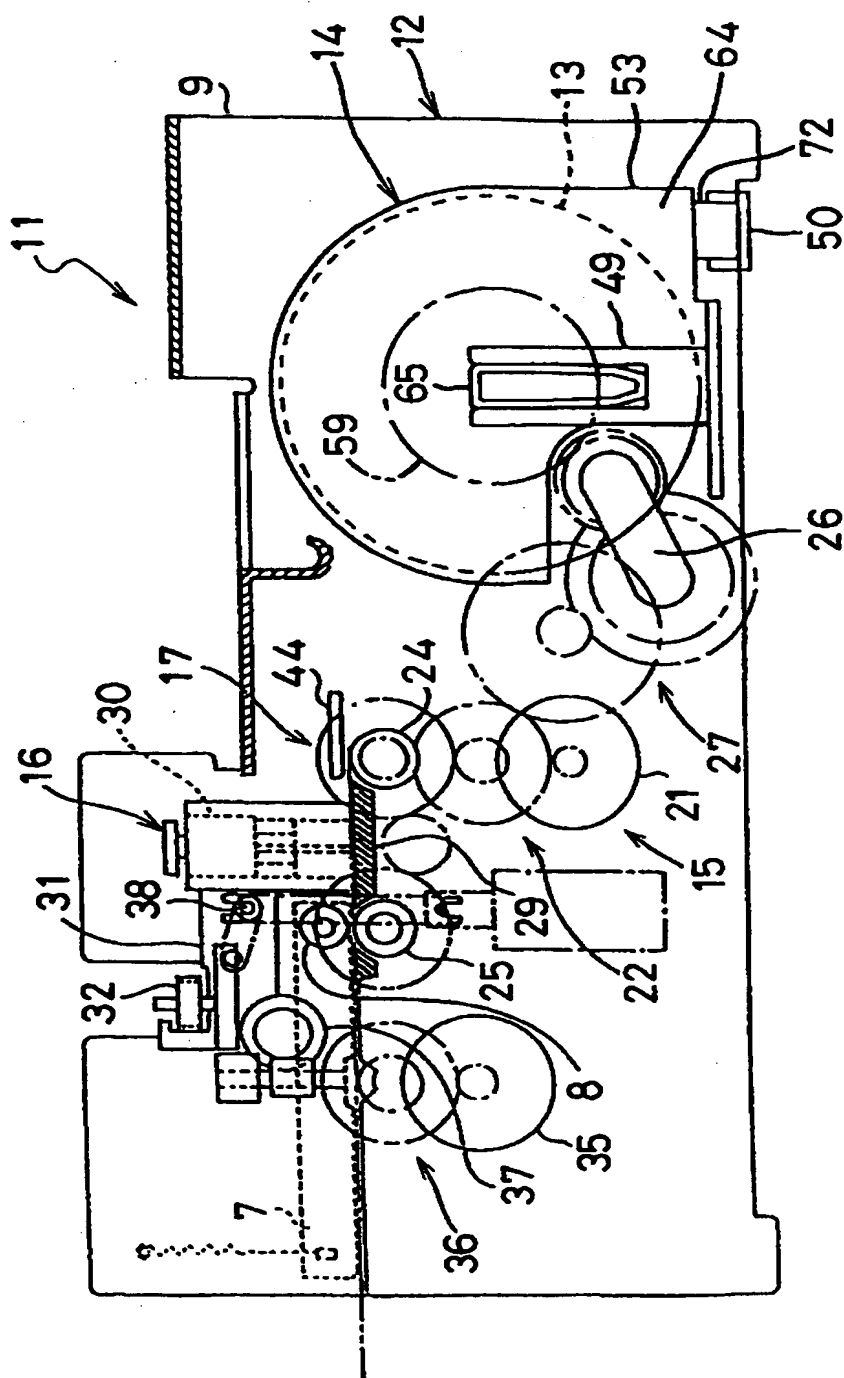
【図 1】



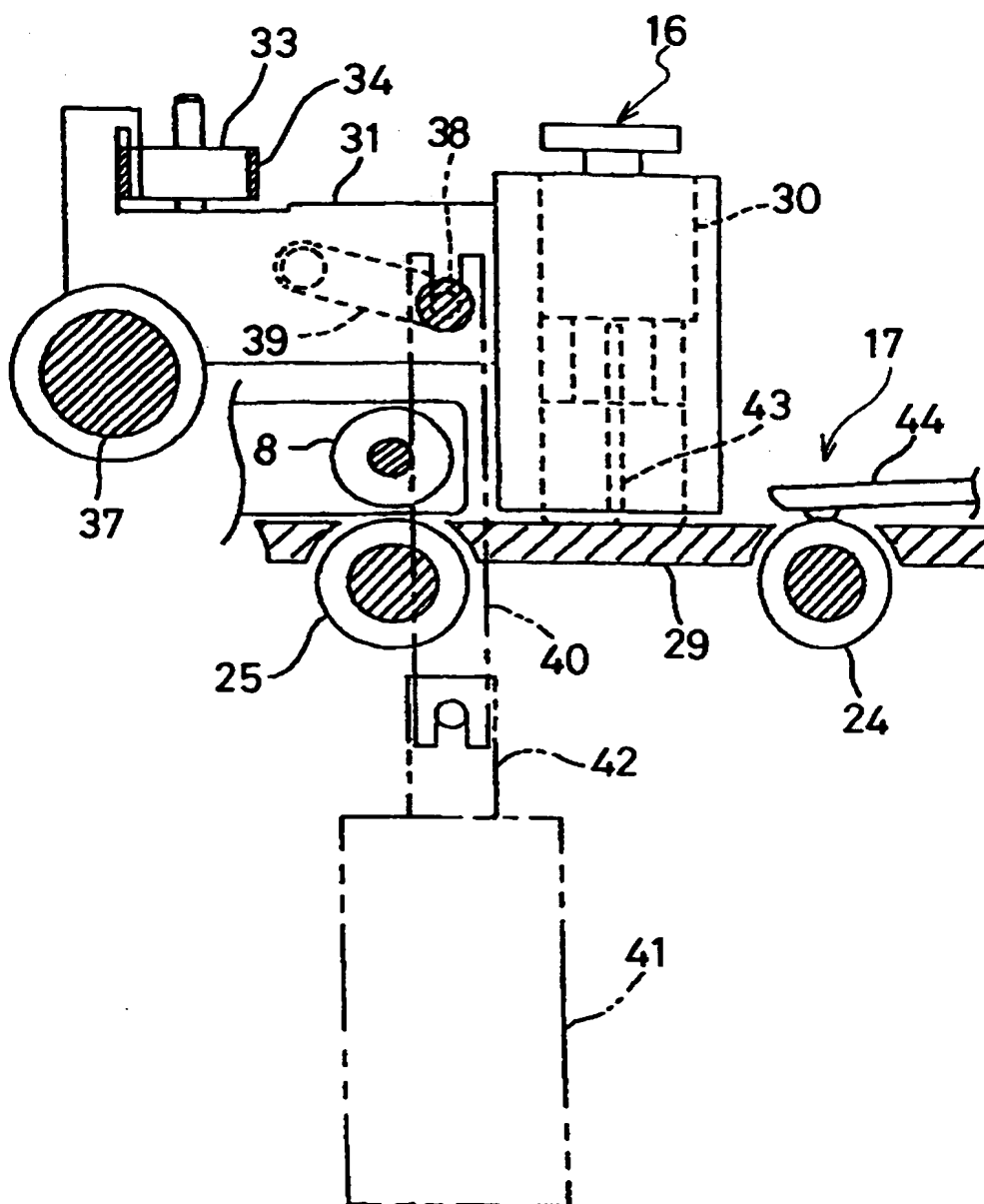
【図 2】



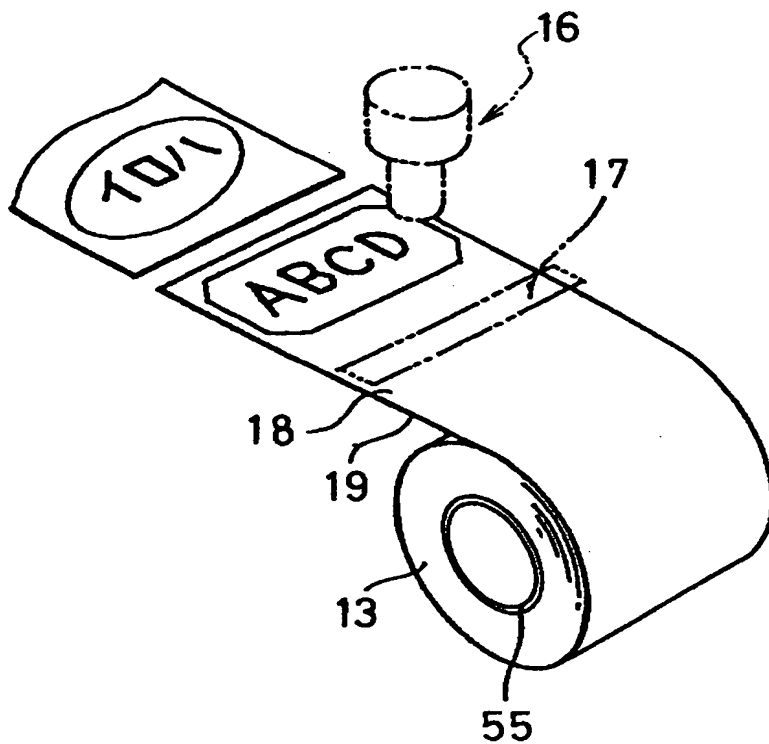
【図 3】



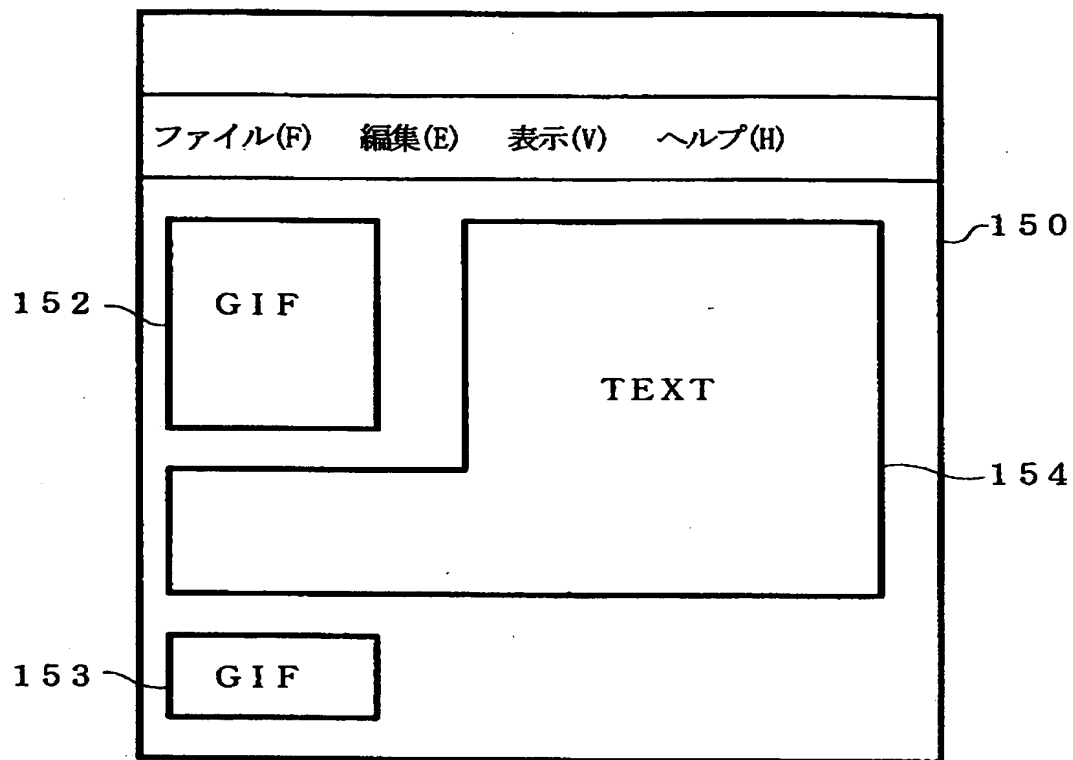
【図 4】



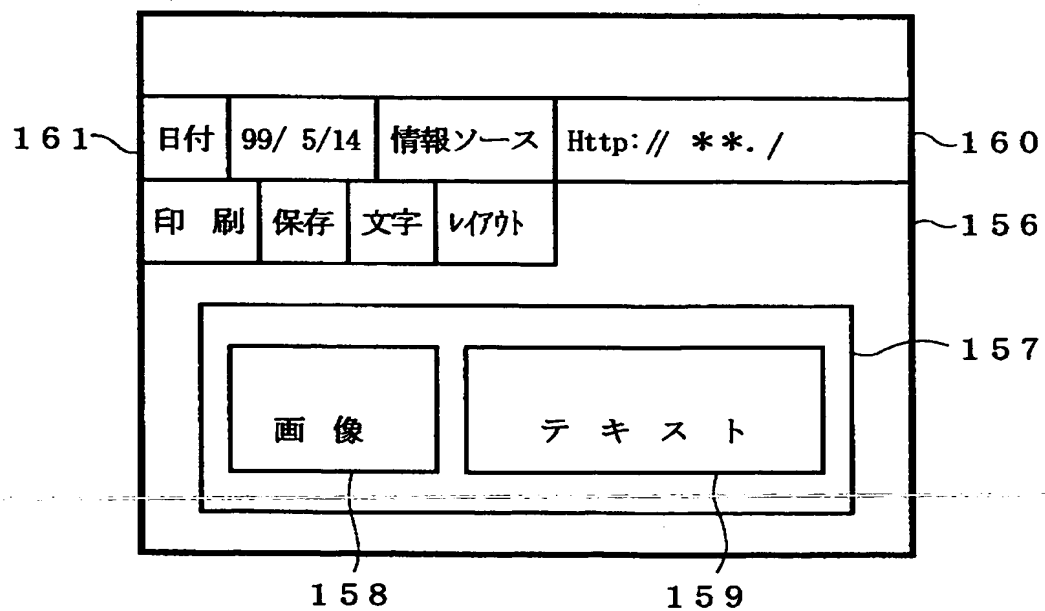
【図 5】



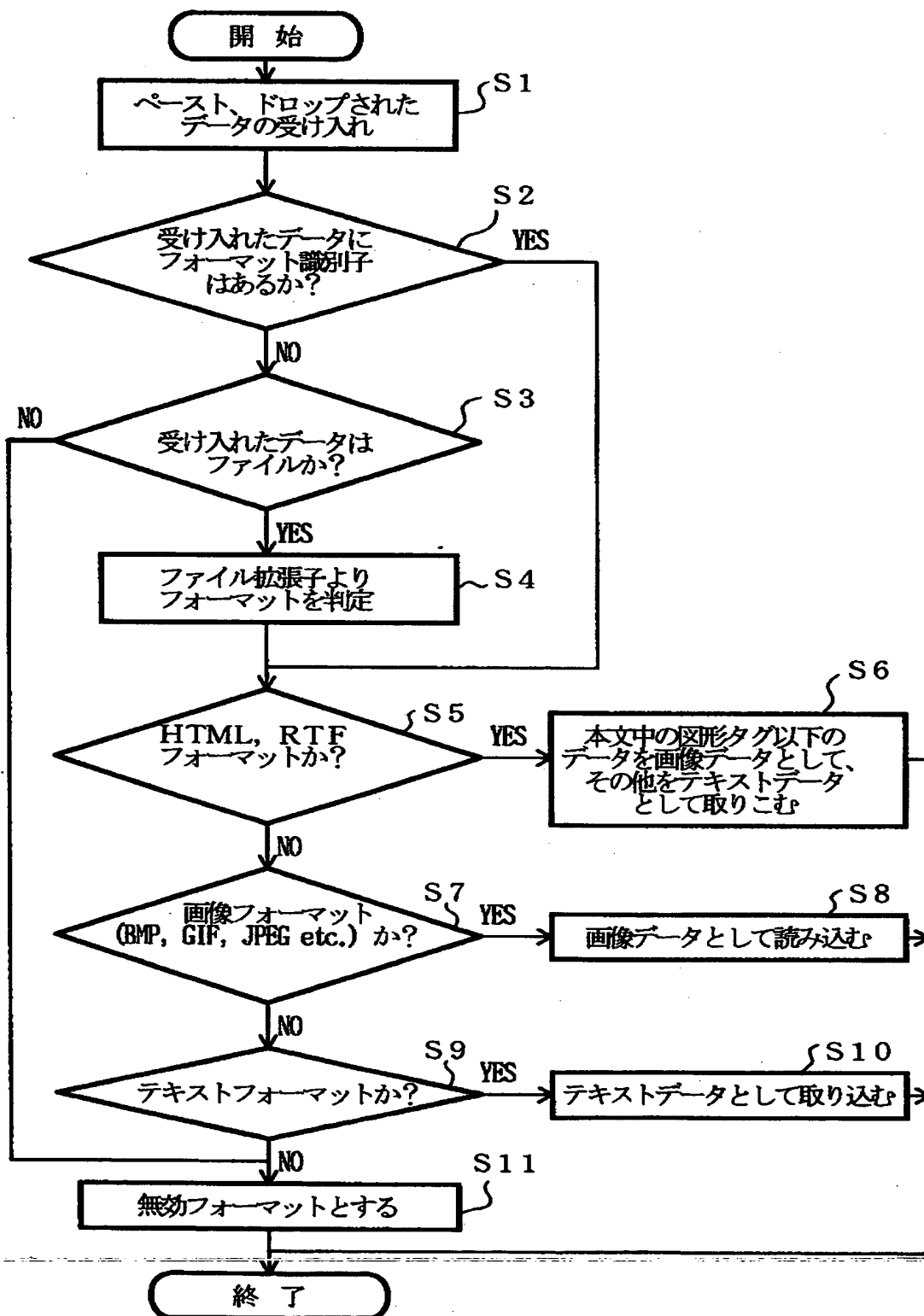
【図 6】



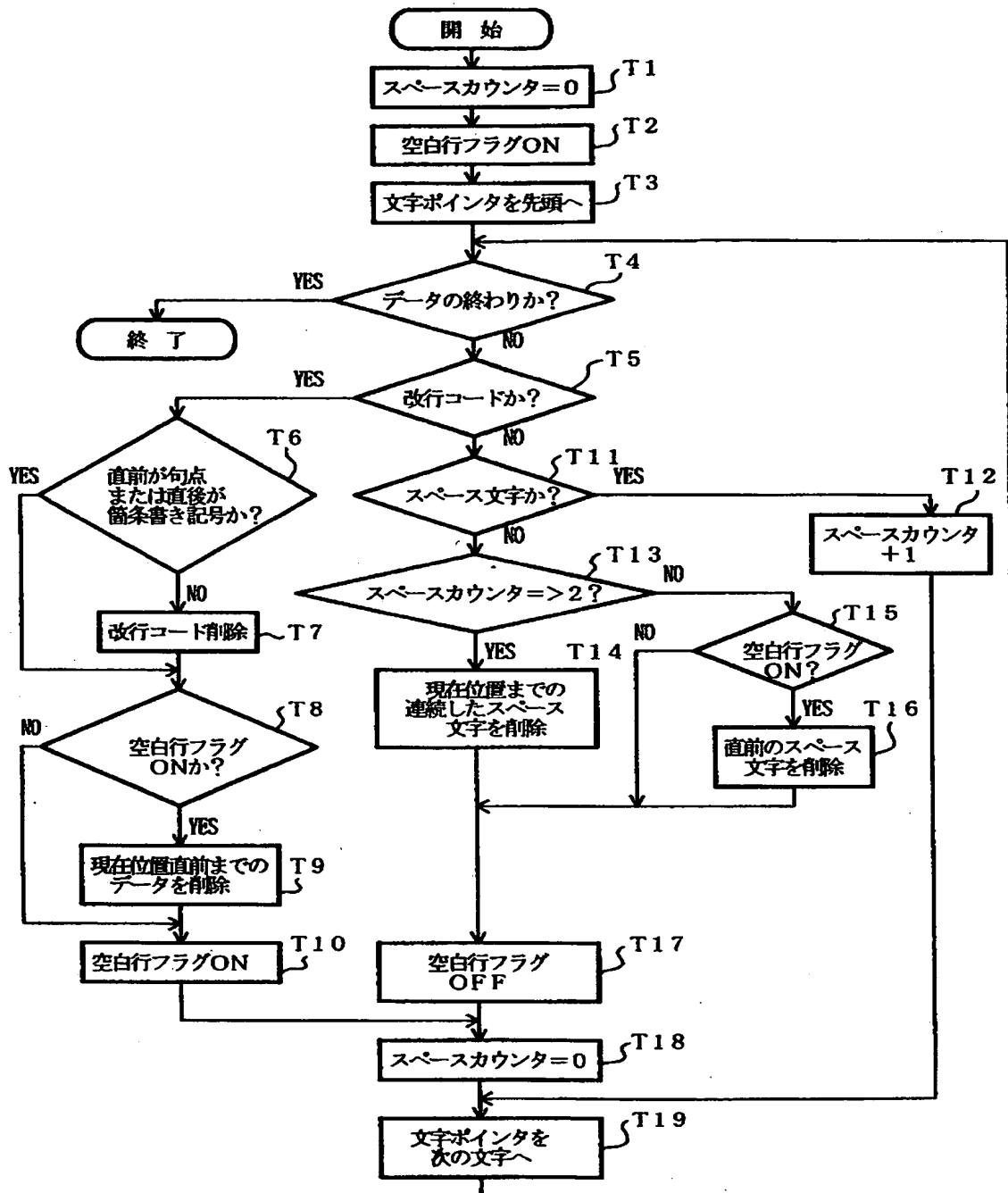
【図 7】



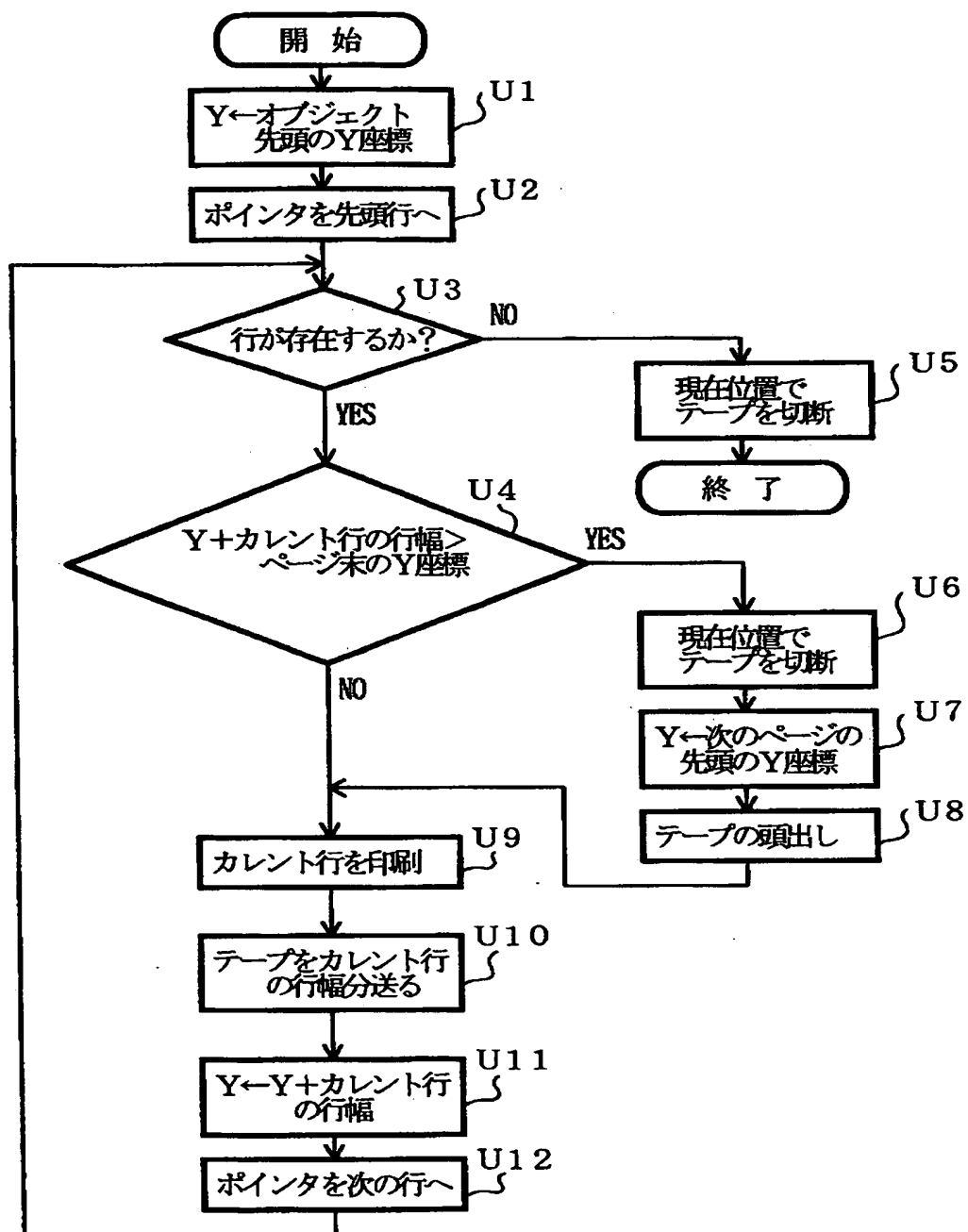
【図 8】



【図 9】



【図 10】



【図 1 1】

1 行目と 2 行目は連続した行なので、途中の改↓
行コードは削除されます。↓
 連続した行で、なおかつインデントした↓
 行に関しては、行をつなげた場合に先頭に↓
 あったスペースが残ってしまうとみっと↓
 もないので、削除する。↓
 ↓
 改行コードが削除されないのは…、↓
 ・ 箇条書き記号「・」などが先頭にある。↓
 ・ 改行コードの直前に句点「。」がある。↓
 の条件を満たす場合である。↓

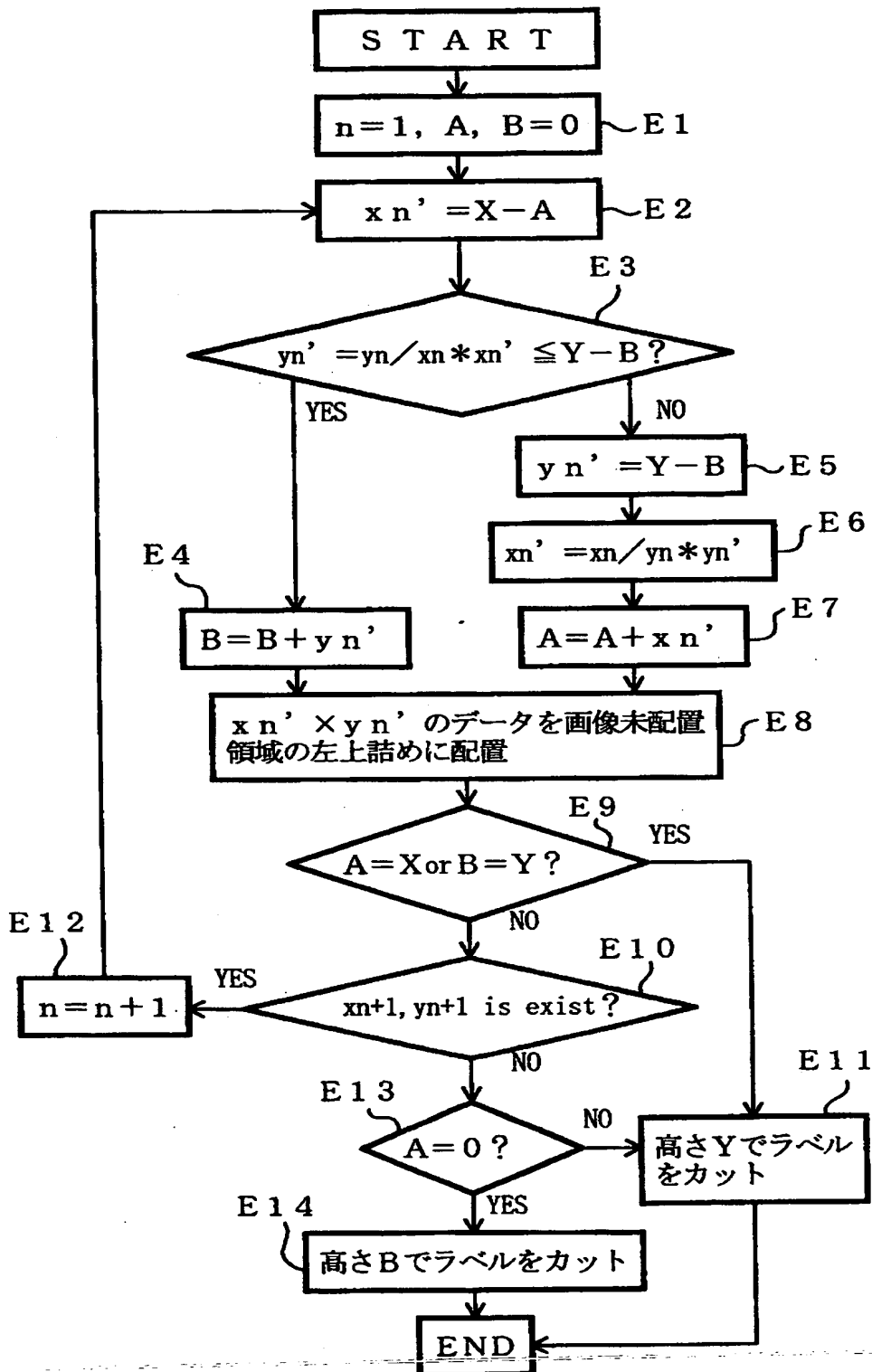
【図 1 2】

1 行目と 2 行目は連続した行
なので、途中の改↓
行コードは削除されます。↓
 連続した行で、なおかつ
 インデントした↓
 行に関しては、行をつなげ
 た場合に先頭に↓
 あったスペースが残って
 しまうとみっと↓
 もないので、削除する。↓
 ↓
 改行コードが削除されないの
は…、↓
 ・ 箇条書き記号「・」などが先頭
 にある。↓
 ・ 改行コードの直前に句点「。」
 がある。↓
 の条件を満たす場合である。↓

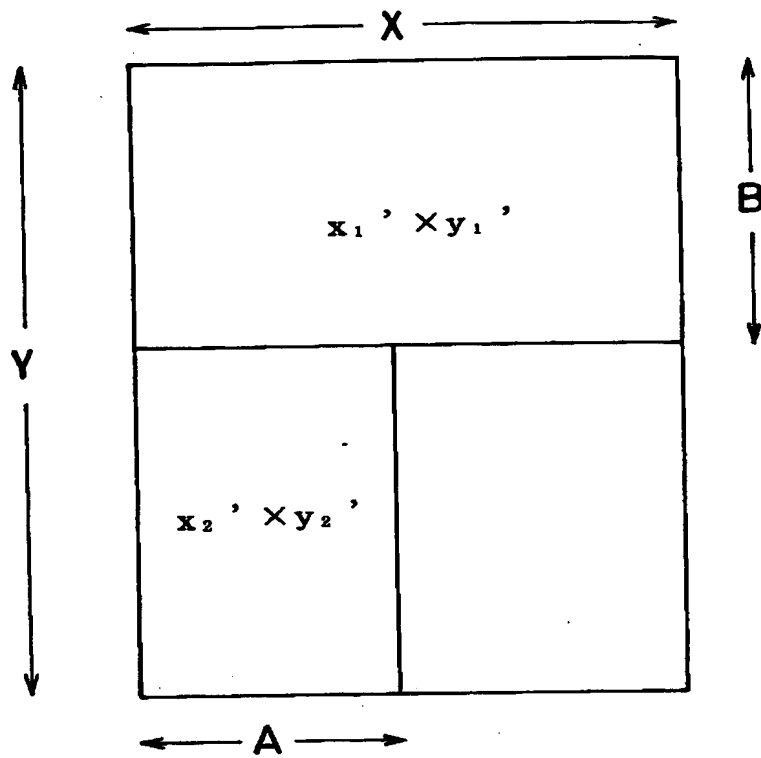
【図 13】

1行目と2行目は連続した行なので、途中の改行コードは削除されます。↓
 連続した行で、なおかつインデントした行に関しては、行をつなげた場合に先頭にあったスペースが残ってしまうとみっともないので削除する。↓
 改行コードが削除されないのは…、↓
 ・箇条書き記号「・」などが先頭にある。↓
 ・改行コードの直前に句点「。」がある。↓
 の条件を満たす場合である。↓

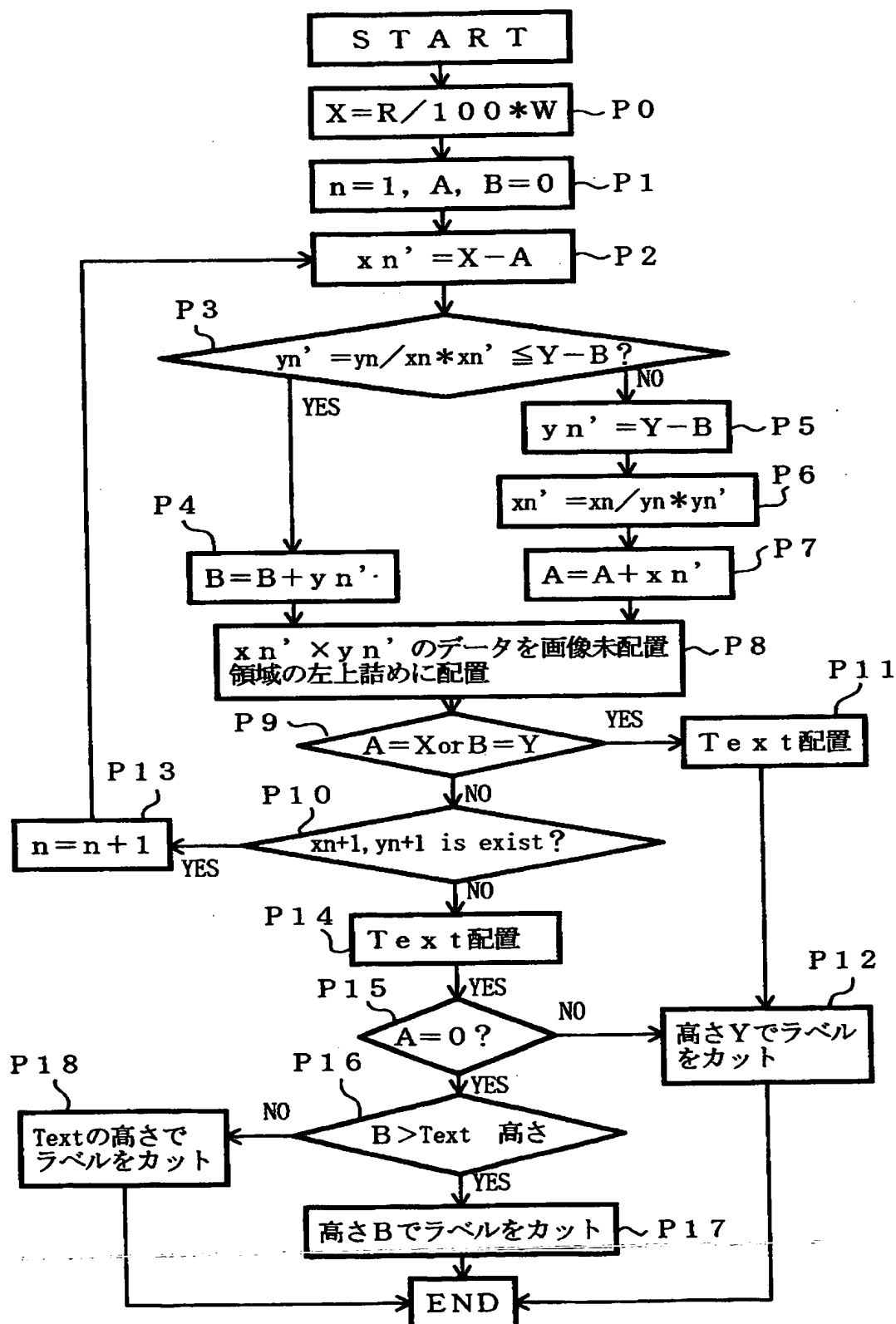
【図 14】



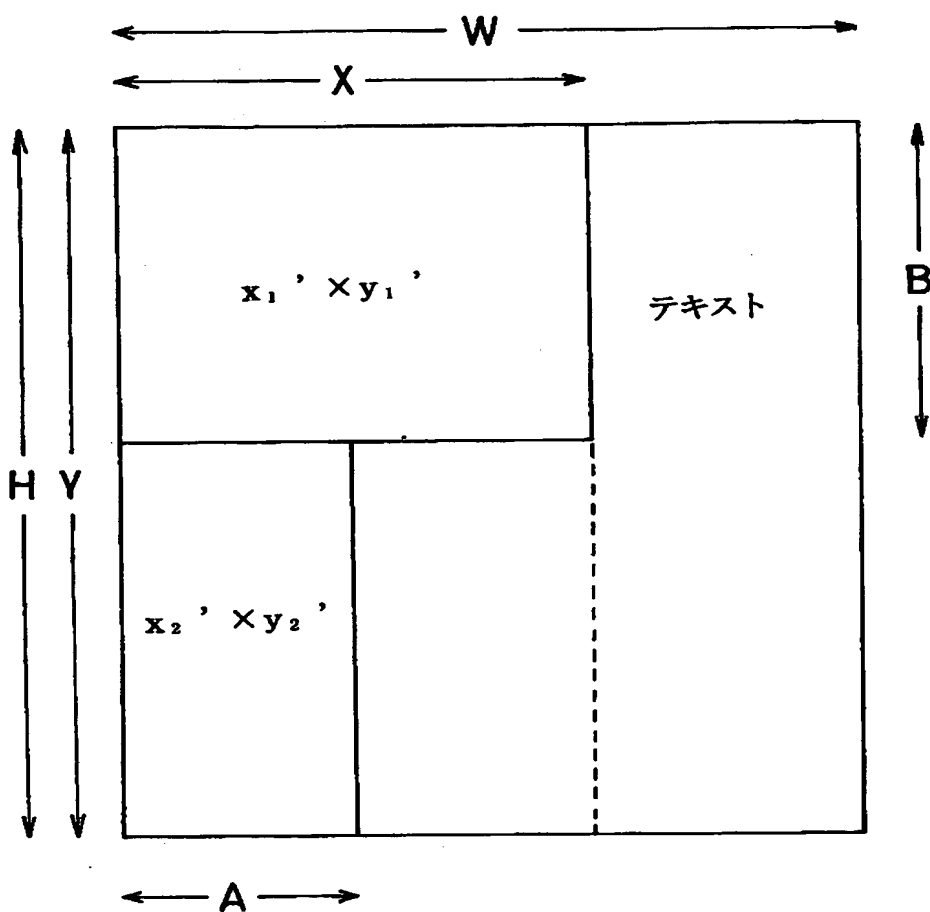
【図 1 5】



【図 16】



【図 1 7】



【書類名】 要約書

【要約】

【課題】 テキストデータを、煩雑な作業を行う必要なく所望サイズの用紙に見やすく印刷させることができるように処理する。

【解決手段】 ブラウザなどから取り込んだテキストデータ内の改行コードを検出し（T 5）、その直前が句点でなく且つ直後が箇条書き記号でない場合に改行コードを削除する（T 6、T 7）。また、テキストデータ内のスペース文字を検出し（T 1 1）、そのときにスペース文字が2以上連続しているときにも（T 1 3）、スペース文字を削除する（T 1 4）。

【選択図】 図 9

出 願 人 履 歴 情 報

識別番号 [000005267]

1. 変更年月日 1990年11月 5日

[変更理由] 住所変更

住 所 愛知県名古屋市瑞穂区苗代町15番1号

氏 名 ブラザー工業株式会社